AD NUMBER

AD480213

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; Oct 1965. Other requests shall be referred to Army Research Office, Research Triangle Park, NC.

AUTHORITY

CFSTI ltr, 22 Apr 1966

480213

//

OPTIMUM POLICIES FOR PARTIALLY OBSERVABLE MARKOV SYSTEMS

by

JAMES STEVEN KAKALIK

Technical Report No. 18

Work Performed Under Contracts

Non-3963 (06), Office of Naval Research
Applications of Probabilistic Models
to Naval Problems
NR 276-004 DSR 9493

and

DA-31-124-ARO-D209, U. S. Army Research Office
Fundamental Investigations in Methods of
Operations Research
DSR 5217

Operations Research Center
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts 02139

October, 1965

Adapted from a thesis, supervised by Professor A. W. Drake, presented to the Department of Electrical Engineering in partial fulfillment of the requirements for the degree of Master of Science, September, 1965. The Computation described in this report was performed at the M.I.T. Computation Center.

ABSTRACT

A partially observable Markov process is a mathematical model of a dynamic probabilistic system which consists of an underlying Markov process obscured from direct observation by imperfect output channels. The observed output R(t) is stochastically related to the underlying state S(t). This model, like the Markov model, is applicable in the analysis of a wide range of sequential decision problems.

The primary area of investigation in this report is the selection of a course of action from a set of alternatives using only the information about the system which is available from the observable outputs. Associated with the model is a cost structure. The decision-maker may use the observed outputs to make inferences about the underlying Markov state and will be assessed rewards or penalties depending on the true state of nature and on the action taken.

The state of knowledge vector $\underline{s(t)}$ summarizes all that is known about the probability of the system being in each of the underlying states as a function of the observed outputs. The optimal policy will specify a course of action to be taken for each possible state of knowledge $\underline{s(t)}$ for all possible t. The policy depends on the decision-maker's knowledge of the underlying Markov state, on the cost structure associated with the model, and on the criterion of optimum used.

Dynamic programming techniques are shown to be of use in the optimization of both transient and steady state policies. The analysis is conducted with the optional availability of a perfect information channel at added cost. Computer programs were written for policy evaluation and optimization, and specific numerical results are included in this report.

FOREWARD

The Operations Research Center at the Massachusetts Institute of Technology is an interdepartmental activity devoted to graduate training and research in the field of operations research. Its products are books, journal articles, technical reports such as this one, and students trained in the theory and practice of operations research.

The Work of this Center is supported, in part, by government contracts and industrial grants-in-aid. Work reported herein was supported (in part) by the Office of Naval Research and the Army Research Office-Durham under contracts number Nonr-3963 (06), NR 276-004 and number DA-31-124-ARO-D-209, respectively. Reproduction in whole or part is permitted for any purpose of the United States Government.

Philip M. Morse
Director of the Center

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to Professor Alvin W. Drake, whose advice, encouragement, and many suggestions made this report possible.

Special thanks must also be given to my many colleagues at the M. I. T. Operations Research Center, in particular Ed Landis, Ralph Miller, and Paul Schweitzer, with whom I have had many helpful discussions.

The computation described in this report was performed at the M. I. T. Computation Center.

James Steven Kakalik

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

One of the basic problems in science and engineering is the construction of models whose mathematical behavior will approximate the physical behavior of real world systems. In the analysis of certain types of nondeterministic systems, the Markov model has shown itself to be a very useful tool.[1] The "partially observable" Markov model is an extension which takes into account the effect of imperfect observations of the state of the dynamic system.

The concept of "state" is central to modelling. The condition or state of a system may be specified by giving the values of relevant parameters. For example, the state of a gas may be specified by giving its temperature, pressure, and the enclosing volume. The state of a highway toll station may be specified at any given instant by the number of collection booths operating and the number of vehicles in each queue. As time progresses, the parameters vary and the system changes state, thereby exhibiting dynamic behavior. The most general probabilistic system would have the parameters taking a continuous allowable range of values, and would allow the parameters to change at any instant in time. This would require a continuous state and continuous time probabilistic model to describe the system. If $s(t)$ is the state of the system at time t, in general $s(t)$ will depend on the entire history of the system previous to time t.

Thus a statistical description of the future of the system will in general depend on both the present state at time t and the complete history of the system previous to time t.

1.1 The Markov Process

If only knowledge of the present state, and not the entire history, is necessary to allow statistical description of the future of the stochastic system, the process is Markovian. Although this is a severely restrictive assumption, in actual fact many real world systems may be accurately modelled as Markov processes. A few prominent areas of Markov process application are marketing, inventory control, traffic, quality control, equipment replacement, routing, and portfolio investment.

To illustrate what the Markovian assumption entails consider the following example:

A housewife buys groceries at the same store once every week. The store carries two brands of milk, A and B. The state of the system in a given week would be the brand of milk she bought that week. The present week is time n, and the probability she buys brand A at week n+1 given her history of purchases is:

$$P[s(n+1)=A \mid s(n)=i,\ s(n-1)=j,\ \ldots\ s(0)=m]$$

where i, j, ...m are either A or B depending on which brand she bought that week.

The Markovian assumption states that the above probability depends only on which brand she purchased this week.

$$P[s(n+1)=A \mid s(n)=i,\ldots s(0)=m] = P[s(n+1)=A \mid s(n)=i] = p_{iA}(n)$$

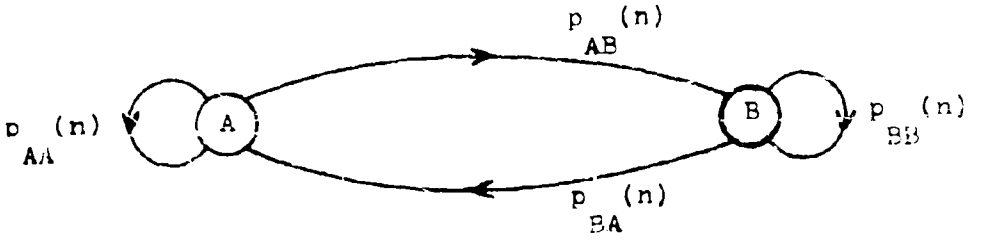


Figure 1 Markov Marketing Model

The transition probability, $p_{ij}(n)$, is the probability that the state at time n will be j if the state at time n-1 was i. The system is called time invariant if $p_{ij}(n)=p_{ij}$ independent of n. For a descrete state and time invariant model with N states, N^2 transition probabilities would be required, not all of which are independent.

1.2 The Partially Observable Process

A partially observable Markov process is one which must be observed through an imperfect output channel. Some examples of imperfect channels are: an imperfect meter, the atmosphere carrying in a signal from outer space, and the incomplete inspection of a manufactured product.

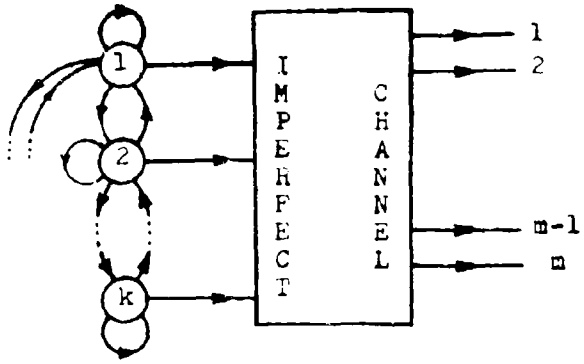


Figure 2 Model of a Partially Observable Markov Process

The model consists of an underlying Markov process which, depending on its true state, supplies values of parameters to the output channels. The imperfect channels operate on the input from the Markov process and yield outputs which in most cases do not allow the observer to ascertain the exact underlying Markov state. In fact, the number of output readings, m, may not even equal the number of true states, k.

The imperfect channel, like the underlying Markov process, is a stochastic process and can be described by the probabilities, $f_{ij}(t)$, which are the probability of output j at time t given that the true state was i.

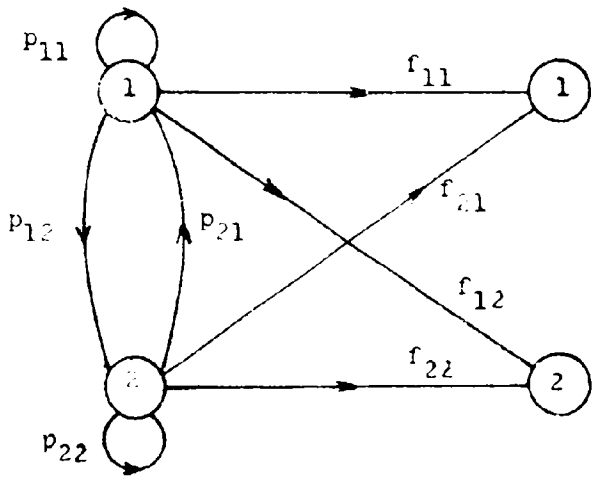


Figure 3 A Two State Model Showing Output Channels

1.3 Reward Structure

Associated with the real life system are decisions and rewards. For example, a decision could be made as to the true Markov state at time t. Various rewards can be defined.

L_{ii}: reward if true state is i and the observer estimates that it is i.

L_{ij}: reward if true state is j and the observer estimates that it is i.

These rewards form the basis for evaluating the effect which a given decision might have.

1.4 Dynamic Inference

The partially observable Markov process is one of a rather large class of systems which consist of one stochastic process monitored through a second stochastic process.

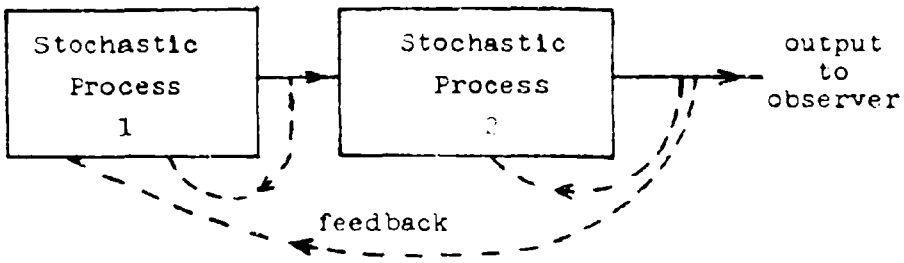


Figure 4 More General "Partially Observable Stochastic Process" Model

In this more general model the process 1 supplies statistical parameters to process 2, which operates on them before presenting observable parameters. Information on variations in the parameters of process 1 must be gleaned from the pattern of the observable parameters output from process 2. The information about process 1 obtained in this manner is then used in decision-making and to predict future developments. The general problem associated with obtaining information about the underlying process is known as "dynamic inference."[5]

Applications of partially observable Markov processes may be found in many areas. For example, the true value of common stocks could be the underlying state with current Wall Street price quotations as the "imperfect" output variable. One might consider the quality of a manufactured product as being the underlying state with results of incomplete inspection supplying the "imperfect" output. Another example, from the marketing area, might consist of

a customer's brand preference as the underlying state, and his latest purchase as the imperfect indicator.

1.5 Previous Investigations

Work has recently been done on various aspects of partially observable Markov processes by Drake,[4] Kramer,[5] and Stoopes[11]. Drake and Kramer discussed formulation of the basic model and considered formation of the $\underline{S}$ vector, or statistical state of knowledge vector, which in essence summarizes all that is known about the probabilities of the underlying Markov process being in each state at time t. They considered methods of updating the $\underline{S}$ vector as new data is received. Drake further considered various decoding schemes on the observed outputs and related errors, as well as information flow and associated costs on simple two state symmetric models.

Stoopes' main investigation was in extending a betting policy formulated by Kelly[7] which entailed betting on various input states a fraction of one's capital proportional to the level of confidence about those input states.

1.6 Statement of Problem

This investigation concerns the optimization of policies associated with physical systems which can be modelled as "partially observable Markov processes." The underlying Markov process can only be observed through a stochastic output channel. Therefore, the future effect of decisions

made utilizing this "imperfect channel" information cannot be stated exactly. Since the observer is dealing with imperfect data, he can only say with probability P_A that the effect of a given decision will be A and with probability P_B the effect of the same decision will be B. This complicates the decision process.

Associated with the selection of a course of action from a set of alternatives is a cost structure. The decision maker may use the observed outputs to make inferences about the underlying Markov state and will be assessed rewards or penalties depending on the true state of nature and on the action taken. The "optimum" policy will depend on the decision-maker's knowledge about the underlying Markov state, on the cost structure associated with the model, and on the criterion of optimum used. There are several possible criteria of a "good" decision. The decision may simply be made so as to maximize the expected value of the reward, or the observer may wish to impose a ceiling on allowed risk and maximize his expected reward while never risking a loss of more than that ceiling. Alternately, some utility function may be imposed upon the rewards and the policy chosen to maximize the expected utility of rewards. In Drake's work, a brief introduction to the above problem is found for a symmetric two state example. This report is a continuation and extension of that introduction.

The major mathematical techniques used for policy optimization are those of dynamic programming which are

covered extensively in Bellman[2], and Bellman and Dreyfus[3]. A dynamic programming algorithm for optimization of regular Markov processes was developed by Howard[9] and extensive work was done in the same area by Schweitzer[10]. In this report, application is found for those techniques in the area of partially observable Markov processes.

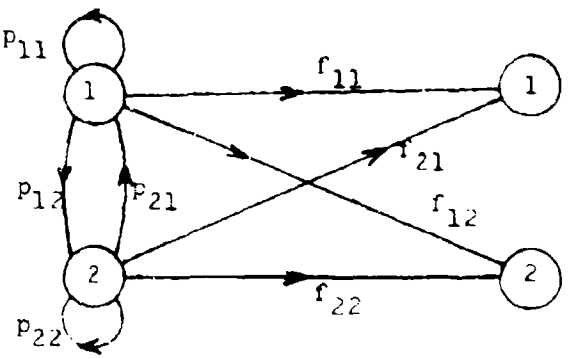
In many practical situations there exists a way to get nearly perfect information about the underlying Markov process--for a price. Therefore the analysis is conducted with the optional availability of a perfect information channel at an additional cost.

A two state Markov process monitored at descrete time intervals by a binary channel will be used to exemplify the ideas presented in this report.

One might give this a physical interpretation from the communications area. Consider that a communications satellite has been placed in orbit and is being used to convey transoceanic messages. Unfortunately because of various interference sources, the satellite may not receive and retransmit an intelligible signal. Therefore the designers built into the satellite a check, whereby the quality of the received message at the satellite is monitored. Then, binary data is transmitted back to the sender at descrete time intervals telling him whether the received message met or did not meet preset standards of quality.

Assume that it has been determined that the process governing whether or not the satellite receives an acceptable signal is essentially Markovian with time invariant

state transition probabilities. The binary signal the monitor returns to the sender is also affected by the interference and is therefore not fully reliable, but the conditional probability distribution of outputs is known. The following partially observable Markov process model is constructed by the decision-maker.



Markov Process generating binary data

outputs from imperfect channel

state 1: acceptable message received at satellite
state 2: unacceptable message received at satellite

Figure 5 Communications Example

The decision-maker can now use this model as an aid in the evaluation of various policies, or courses of action. Using the binary output data, inferences can be made about the signal quality at the satellite. The knowledge about signal quality can then be used along with the cost structure to evaluate the expected consequences of various courses of action.

Various alternatives might be available to the decision-maker. He might continue regular transmission, resend a portion of the message, discontinue transmission for one or more time units, conduct additional tests of signal quality, or build a new and different communications system.

Markovian models have shown themselves to be very useful in the past. The techniques developed in this report allow the extension of analytical methods for optimal decision making to include the case of the Markov process being "obscured" by an imperfect information channel.

CHAPTER II

OPTIMAL TIME-DEPENDENT POLICIES

When making a decision, an extremely useful quantity to know is the total reward or cost that can be expected as a consequence of the particular decision made. Dynamic programming allows the calculation of future expected utility of rewards as a function of policy in sequential decision problems, and therefore allows the selection of a decision to maximize total expected utility of rewards.

In sequential decision problems, decisions may be made at certain points in time and each decision will, in general, carry with it implications which extend far into the future and affect decisions as yet unmade. Likewise, what the policy-maker intends to do in the future will affect his present decision.

There are two basic techniques in dynamic programming. These involve solving a problem in either "value space" or "policy space." In this chapter the "value space" technique is explained and is applied to partially observable Markov processes. In Chapter III the "policy space" technique will be employed in the determination of optimum policies.

Consider the two state communications example from Chapter I where the observer periodically receives information on the reception quality at the satellite.

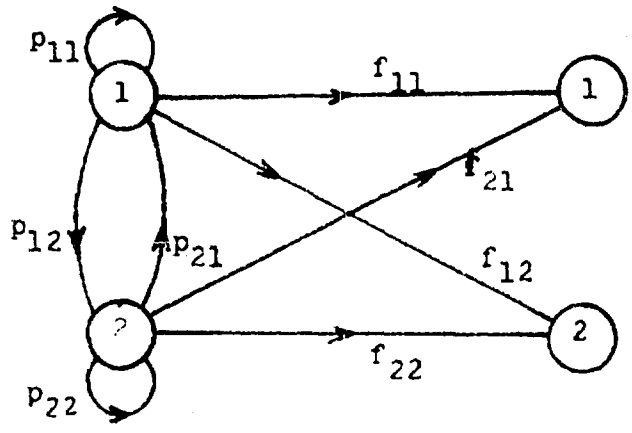


Figure 6 Communications Example

Markov state 1: message at satellite meets preset standards

Markov state 2: message at satellite fails to meet preset standards

Output 1: Ground Observer receives signal--"state 1"

Output 2: Ground Observer receives signal--"state 2"

The ground observer now has to make a decision on the basis of the stochastically inaccurate output signals. Assume first that he has only two options open:

1) Continue transmitting until next output is received
2) Stop transmitting and check again one time unit later

This problem will be first approached in the time-dependent case. That is, the observer doesn't have

unlimited access to the use of the satellite but must definitely quit at some time n units into the future. It will be convenient to measure one unit of time as the time between output signals. When dynamic programming techniques are applied in the analysis of processes which will terminate at some specific time in the future, it is conventional to call the termination time zero and measure time in reverse of normal order. Thus, in this example, the current time is n, and the process must terminate at time zero which is n time units into the future. The time independent policy, where the physical process terminates far into the future or continues indefinitely, will be considered in Chapter III.

Inherent in the decision problem is a reward structure.

L_{11}: Utility of reward if he continues transmitting and the underlying Markov state is 1

L_{22}: Utility of reward if he stops transmitting and the underlying Markov state is 2

L_{12}: Utility of reward if he continues transmitting and the underlying Markov state is 2

L_{21}: Utility of reward if he stops transmitting and the underlying Markov state is 1.

The preceding problem uses a two state process with two options allowed at each decision point. That type problem will now be solved in general.

2.1 Notation

In computations to come the following shorthand notation will be useful:

$S(n)=x$: Underlying Markov state at time n is x

$R(n)$: Output response at time n

To summarize the decision-makers's knowledge at time n:

$$s_x(n) = P[S(n)=x \mid R(n), R(n+1), R(n+2)...]$$

= probability that the underlying Markov state is x at time n, given the past history of observed outputs.

For an N state process, a "state of knowledge" vector is defined:

$$\underline{s(n)} = \underline{s_1(n),\ s_2(n),\ \ldots\ s_N(n)}$$

The "state of knowledge" vector summarizes all that the observer knows about the process at time n. For a two state process, $s_1(n)$ is sufficient to determine the $\underline{s(n)}$ vector because it is known that the underlying Markov state is either 1 or 2 and therefore $s_2(n)$ can be found from $s_1(n)$.

$$s_2(n) = 1-s_1(n)$$

$$P_x[\underline{s(n)}] = P[R(n-1) = x \mid \underline{s(n)}]$$

= probability that the next output is x given the current state of knowledge

The state of knowledge vector must be updated as new information is received:

$\underline{T_x[\underline{s(n)}]}$ = updated state of knowledge at time n-1 given that the output at time n-1 was x. The new state of knowledge is a function of the old $\underline{s(n)}$ and the decision-maker sees R(n-1) before he must update $\underline{s(n)}$.

2.2 Prediction of Outputs

The probability distribution on the next output to be received will prove useful. The state of knowledge vector gives a probability distribution on the underlying Markov state, and if the underlying Markov state were know to be i, the probability of output j would be f_{ij}. For the two state case, the next output at time n-1 is predicted to be 1 or 2 with probabilities:

$$P_1[\underline{s(n)}] = s_1(n)\,(p_{11}f_{11} + p_{12}f_{21}) + s_2(n)\,(p_{22}f_{21} + p_{21}f_{11})$$

$$P_2[\underline{s(n)}] = s_1(n)\,(p_{11}f_{12} + p_{12}f_{22}) + s_2(n)\,(p_{22}f_{22} + p_{21}f_{12})$$

The above equations can be written in matrix form for the two state case and then extended to the N state case.

Two state process:

$$[P] = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \qquad [F] = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$$

$\underline{P[\underline{s(n)}]}$ = row matrix of probabilities of next output reading

$$= \underline{s(n)}\,[P]\,[F] = \underline{P_1[\underline{s(n)}],\ P_2[\underline{s(n)}]}$$

N state process:

$$[P] = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & \cdot & & \\ \vdots & & \ddots & \\ p_{N1} & & & p_{NN} \end{bmatrix} \qquad [F] = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & \cdot & & \\ \vdots & & \ddots & \\ f_{N1} & & & f_{NN} \end{bmatrix}$$

$$\underline{P[s(n)]} = \underline{s(n)}\,[P]\,[F] = \underline{P_1[s(n)],\ P_2[s(n)],\ldots,P_N[s(n)]}$$

If $A_i]$ is defined as the i^{th} column of matrix [A], the components of the $\underline{P[s(n)]}$ vector are simply written.

$$P_i[\underline{s(n)}] = \underline{s(n)}\,[P]\,F_i] \quad = \quad \text{a scalar}$$

2.3 Updating the State of Knowledge

The state of knowledge vector changes with time and it will be necessary to update it as new information is received. Recall that time is to be measured in reverse order. The current time is n and the process must terminate at time zero which is n time units into the future. If the decision-maker does not have the output readings available, his state of knowledge vector would change with time as follows:

$\underline{\psi(n-1)}$ = new state of knowledge vector if the decision-maker does not have the output readings available.

For the 2 state process:

$$\underline{\psi(n-1)} = \underline{s(n)}\,[P] = \underline{\psi_1(n-1),\ \psi_2(n-1)}$$

$$\psi_1(n-1) = s_1(n)p_{11} + s_2(n)p_{21}$$

$$\psi_2(n-1) = s_1(n)p_{12} + s_2(n)p_{22}$$

For the N state process the matrix equation is the same.

$$\underline{\psi(n-1)} = \underline{s(n)}[P] = \underline{\psi_1(n-1),\ \psi_2(n-1),\ldots,\psi_N(n-1)}$$

Now consider giving the decision-maker the advantage of seeing the output response R(n-1) before he must update the state of knowledge vector. Given that output "i" has been observed, the new state of knowledge vector can be computed to be:

$$\underline{T_i[s(n)]} = [\underline{s(n-1)} \text{ given } R(n-1)=i] = \underline{T_{i,1}[s(n)],\ T_{i,2}[s(n)]}$$

Given output "i", the $j^{\underline{th}}$ component of the new state of knowledge vector is:

$$T_{i,j}[\underline{s(n)}] = P[S(n-1)=j \mid R(n-1)=i \text{ and } \underline{s(n)}]$$

$$= \frac{P[S(n-1)=j \text{ and } R(n-1)=i \mid \underline{s(n)}]}{P[R(n-1)=i \mid \underline{s(n)}]}$$

$$T_{i,j}[\underline{s(n)}] = \frac{P[S(n-1)=j \text{ and } R(n-1)=i \mid \underline{s(n)}]}{P_i[\underline{s(n)}]}$$

$$= \frac{P[R(n-1)=i \mid S(n-1)=j \text{ and } \underline{s(n)}]\ P[S(n-1)=j \mid \underline{s(n)}]}{P_i[\underline{s(n)}]}$$

$$= \frac{f_{ji}\ \psi_j(n-1)}{P_i[\underline{s(n)}]}$$

For a two state process, using the above relation:

$$T_{1,1}[\underline{s(n)}] = \frac{[s_1(n)p_{11} + s_2(n)p_{21}]\ f_{11}}{P_1[\underline{s(n)}]}$$

$$T_{1,2}[\underline{s(n)}] = 1-T_{1,1}[\underline{s(n)}] = \frac{[s_1(n)p_{12} + s_2(n)p_{22}]\ f_{21}}{P_1[\underline{s(n)}]}$$

$$T_{2,1}[\underline{s(n)}] = \frac{[s_1(n)p_{11} + s_2(n)p_{21}]\ f_{12}}{P_2[\underline{s(n)}]}$$

$$T_{2,2}[\underline{s(n)}] = 1-T_{2,1}[\underline{s(n)}] = \frac{[s_1(n)p_{12} + s_2(n)p_{22}]\ f_{22}}{P_2[\underline{s(n)}]}$$

The new state of knowledge vector $\underline{s(n-1)}$ will then depend on what output is observed at time n-1.

If the output at time n-1 was 1:

$$\underline{s(n-1)} = T_1[\underline{s(n)}] = \underline{T_{1,1}[s(n)],\ T_{1,2}[s(n)]}$$

If the output at time n-1 was 2:

$$\underline{s(n-1)} = T_2[\underline{s(n)}] = \underline{T_{2,1}[s(n)],\ T_{2,2}[s(n)]}$$

The observer will receive an output reading "i" and must update his state of knowledge $\underline{s(n-1)} = T_i[\underline{s(n)}]$. Even if no output reading is available to the decision-maker, his state of knowledge still changes $\underline{s(n-1)} = \underline{\psi(n-1)}$

2.4 Policy Evaluation

If there are a number of options available to the decision-maker for each value of the state of knowledge vector $\underline{s(n)}$, then a policy would specify which option (k) to take for each possible $\underline{s(n)}$ for all n. The choice of policies will usually be affected by the total expected reward associated with each different policy.

The expected reward can be separated into two categories: immediate and future. The reward to be expected during the current time unit only is known as immediate reward. Future reward is the expected reward in the aggregate of all future time.

This grouping of rewards forms the basis of the dynamic programming equations to be used throughout the remainder of this report.

$$\begin{pmatrix}\text{Total expected reward}\\ \text{with } n \text{ time}\\ \text{units remaining}\end{pmatrix} = \begin{pmatrix}\text{Immediate expected}\\ \text{reward in the}\\ \text{current unit of time}\end{pmatrix}$$

$$+ \begin{pmatrix}\text{Total expected future}\\ \text{reward with n-1}\\ \text{time remaining}\end{pmatrix}$$

.

The expected immediate reward will be affected by the option (k) which the decision-maker exercizes at time n and by his state of knowledge.

$q_k[\underline{s(n)}]$ = Immediate expected reward in the current unit of time as a function of the state of knowledge if option k is exercized.

Continuing with the two state process, recall that at present only two options are allowed. In general terms those two options are:

k=1 : estimate underlying Markov state 1 as the current Markov state and act accordingly

k=2 : estimate underlying Markov state 2 as the current Markov state and act accordingly

For the communications example, these options are:

k=1 : continue transmitting until the next output is received

k=2 : stop transmitting and check again one time unit later

The immediate expected earnings in the current unit of time are:

$$q_k[\underline{s(n)}] = s_1(n)L_{k1} + s_2(n)L_{k2} = \sum_{j=1}^{2} s_j(n)L_{kj}$$

where L_{kj} = earnings if estimate k and true state is j

The total expected earnings are given by the expression below for a fixed policy.

$F^n[\underline{s(n)}]$ = total expected earnings with n time left as a function of the current state of knowledge.

$$F^n[\underline{s(n)}] = \begin{pmatrix}\text{Immediate}\\ \text{expected}\\ \text{earnings}\end{pmatrix} + \begin{pmatrix}\text{Expected earnings}\\ \text{in time n-1 given}\\ R(n-1) = 1\end{pmatrix} \cdot P_1[\underline{s(n)}]$$

$$+ \begin{pmatrix}\text{Expected earnings}\\ \text{with time n-1}\\ \text{given } R(n-1)=2\end{pmatrix} \cdot P_2[\underline{s(n)}]$$

$$F^n[\underline{s(n)}] = q_k[\underline{s(n)}] + P_1[\underline{s(n)}]\, F^{n-1}[\underline{T_1(s(n))}]$$

$$+ P_2[\underline{s(n)}]\, F^{n-1}[\underline{T_2(s(n))}]$$

It is possible to solve this functional equation for the total expected reward and thus estimate the effect of a given policy choice.

2.5 Policy Optimization

To optimize the decision, a criterion of "optimum"

must first be chosen. The rewards earned may be in the form of money, time, material goods, etc. and the reward, L_{ij}, is really a utility or index of usefulness to the decision-maker. If "optimum" means maximizing the total expected reward, or total expected utility, then dynamic programming will yield an optimum policy by solving the functional equation below subject to the initial conditions $F^0[\underline{s(0)}]$ specified by the decision-maker.

$$F^n[\underline{s(n)}] = \underset{k}{\text{maximum}} \left\{ q_k[\underline{s(n)}] + P_1[\underline{s(n)}]F^{n-1}[\underline{T_1(s(n))}] + P_2[\underline{s(n)}]\, F^{n-1}[\underline{T_2(s(n))}] \right\}$$

where k represents the options available at time n.

Bellman's[1] principle of optimality states that the computed solution will in fact be the best policy based on the criterion of maximizing expected utility.

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

In solving the functional equations Bellman's principle is used in the following manner. Subject to the initial states and decisions $F^0[\underline{s(0)}]$ is specified. Then $F^1[\underline{s(1)}]$ is found using the functional equation relation and the

values $F^0[\underline{s(0)}]$. $F^2[\underline{s(1)}]$ is then found from $F^1[\underline{s(1)}]$ and so on. Each time the decision at time n is made consistent with decisions already made.

2.6 Perfect Information

In many practical situations, perfect or nearly perfect information about the underlying Markov state can be obtained at increased expense. With this perfect information channel at a net cost of an additional "A" dollars, there is a third option open.

If the perfect information channel is used at time n the state of knowledge vector becomes:

$$\underline{s(n)} \longrightarrow \underline{1\ ,\ 0} \quad \text{with probability } s_1(n)$$
$$\underline{s(n)} \longrightarrow \underline{0\ ,\ 1} \quad \text{with probability } s_2(n)$$

Using option 3 at time n, the associated total expected reward before the channel is used is given below.

$$F^n[\underline{s(n)}] = \begin{pmatrix} -A + s_1(n)L_{11} + s_2(n)L_{22} \\ + s_1(n)\ P_1[\underline{1,0}]F^{n-1}[\underline{T_1(1,0)}] \\ + s_1(n)\ P_2[\underline{1,0}]F^{n-1}[\underline{T_2(1,0)}] \\ + s_2(n)\ P_1[\underline{0,1}]F^{n-1}[\underline{T_1(0,1)}] \\ + s_2(n)\ P_2[\underline{0,1}]F^{n-1}[\underline{T_2(0,1)}] \end{pmatrix}$$

Therefore, the equations to be solved for the optimum policy are:

$$F^n[\underline{s(n)}] = \max_k \begin{cases} \underline{k=1}: & \text{expected reward if option 1 is exercized} \\ \underline{k=2}: & \text{expected reward if option 2 is exercized} \\ \underline{k=3}: & \text{expected reward if perfect information channel is used} \end{cases}$$

$$F^n[\underline{s(n)}] = \max_k \begin{cases} \underline{k=1}: & s_1(n)L_{11} + s_2(n)L_{12} \\ & + P_1[\underline{s(n)}]F^{n-1}[\underline{T_1[s(n)]}] \\ & + P_2[\underline{s(n)}]F^{n-1}[\underline{T_2[s(n)]}] \\ \underline{k=2}: & s_1(n)L_{21} + s_2(n)L_{22} \\ & + P_1[\underline{s(n)}]F^{n-1}[\underline{T_1[s(n)]}] \\ & + P_2[\underline{s(n)}]F^{n-1}[\underline{T_2[s(n)]}] \\ \underline{k=3}: & -A + s_1(n)L_{11} + s_2(n)L_{22} \\ & + s_1(n)\ P_1[\underline{1,0}]F^{n-1}[\underline{T_1[1,0]}] \\ & + s_1(n)\ P_2[\underline{1,0}]F^{n-1}[\underline{T_2[1,0]}] \\ & + s_2(n)\ P_1[\underline{0,1}]F^{n-1}[\underline{T_1[0,1]}] \\ & + s_2(n)\ P_2[\underline{0,1}]F^{n-1}[\underline{T_2[0,1]}] \end{cases}$$

Comparing options 1 and 2, the decision-maker will choose option 1 over option 2, i.e. estimate the underlying Markov state as 1 instead of 2 and act accordingly if:

$$\begin{pmatrix}\text{option 1 expected}\\ \text{reward}\end{pmatrix} \geq \begin{pmatrix}\text{option 2 expected}\\ \text{reward}\end{pmatrix}$$

$$s_1(n)L_{11} + s_2(n)L_{12} \geq s_1(n)L_{21} + s_2(n)L_{22}$$

$$s_1(n)L_{11} + [1-s_1(n)]L_{12} \geq s_1(n)L_{21} + [1-s_1(n)]L_{22}$$

$$s_1(n) \geq \frac{L_{22} - L_{12}}{L_{11} + L_{22} - L_{12} - L_{21}}$$

Therefore if only options 1 and 2 are available the solution is trivial and made on the basis of highest immediate expected return, $q[\underline{s(n)}]$. Adding option 3 has the effect of allowing him to "invest" A dollars now, in hope of getting higher overall future returns.

Comparing options 1 and 3, the decision-maker will estimate 1 instead of using the perfect channel (option 3) if:

$$\begin{pmatrix} s_1(n)L_{11}+s_2(n)L_{12} \\ + P_1[\underline{s(n)}]F^{n-1}[\underline{T_1[\underline{s(n)}]}] \\ + P_2[\underline{s(n)}]F^{n-1}[\underline{T_2[\underline{s(n)}]}] \end{pmatrix} \geq \begin{pmatrix} -A + s_1(n)L_{11} + s_2(n)L_{22} \\ + s_1(n)\ P_1[\underline{1,0}]F^{n-1}[\underline{T_1[\underline{1,0}]}] \\ + s_1(n)\ P_2[\underline{1,0}]F^{n-1}[\underline{T_2[\underline{1,0}]}] \\ + s_2(n)\ P_1[\underline{0,1}]F^{n-1}[\underline{T_1[\underline{0,1}]}] \\ + s_2(n)\ P_2[\underline{0,1}]F^{n-1}[\underline{T_2[\underline{0,1}]}] \end{pmatrix}$$

The equality condition above can be interpreted as the value of $\underline{s(n)}$ for which the decision-maker is indifferent

between options 1 and 3. If $s_1^*(n)$ is defined as the value of $s_1(n)$ for which the decision-maker is indifferent between option 1 and 3, the preceding equation can be solved for $s_1^*(n)$ and then option 1 will be chosen over option 3 if $s_1(n) \geq s_1^*(n)$.

Similarly, comparing options 2 and 3, option 2 will be preferred over option 3 if $s_1(n) \leq s_1^{**}(n)$.

Summarizing, using the criterion of maximum expected utility of rewards, the optimal decision will be:

a) estimate state 1 and act accordingly if $s_1(n) \geq \frac{L_{22}-L_{12}}{L_{11}+L_{22}-L_{12}-L_{21}}$

and $s_1(n) \geq s_1^*(n)$

b) estimate state 2 and act accordingly if $s_1(n) \leq \frac{L_{22}-L_{12}}{L_{11}+L_{22}-L_{12}-L_{21}}$

and $s_1(n) \leq s_1^{**}(n)$

c) reascertain true state using perfect channel if $\underline{s(n)}$ doesn't satisfy either a) or b).

2.7 Additional Partial Information

A further practical generalization can be made by assuming that the observer has a choice of using a second imperfect channel which is better than the first, costs "B" dollars more per usage, but still isn't perfect. The decision-maker has also to decide now whether the better channel is worth the extra money for any possible state

of knowledge and time (n).

To answer this question, he adds a fourth option to his functional equation. The new [P'] matrix corresponding to the new output channel affects the updating of the state of knowledge if the new channel is used. Therefore, define a new quantity $\underline{T_1'(\cdot)}$ to represent the new state of knowledge after the output reading "1" from the new channel has been considered.

Under option 4 at time n:

$$F^n[\underline{s(n)}] = -B + s_1(n) \cdot \begin{pmatrix}\text{expected earnings in time n if indi-}\\ \text{cation 1 is received from the new}\\ \text{channel}\end{pmatrix} + s_2(n) \cdot \begin{pmatrix}\text{expected earnings in time n if indi-}\\ \text{cation 2 is received from the new}\\ \text{channel}\end{pmatrix}$$

$$F^n[\underline{s(n)}] = -B + s_1(n)X + s_2(n)Y$$

where:

$$X = q_4[\underline{T_1'[s(n)]}] + P_1[\underline{T_1'[s(n)]}]F^{n-1}\{T_1[\underline{T_1'[s(n)]}]\} + P_2[\underline{T_1'[s(n)]}]F^{n-1}\{T_2[\underline{T_1'[s(n)]}]\}$$

$$Y = q_4[\underline{T_2'[s(n)]}] + P_1[\underline{T_2'[s(n)]}]F^{n-1}\{T_1[\underline{T_2'[s(n)]}]\} + P_2[\underline{T_2'[s(n)]}]F^{n-1}\{T_2[\underline{T_2'[s(n)]}]\}$$

2.6 Example

The value of this technique is that a solution may be found for any length of time, n, remaining. However, this becomes impractical as n becomes large. Fortunately the equations will converge on an optimal policy which, for large n, is independent of n. This "steady state" policy may become discernible for very small n in some problems. The next chapter presents a method for obtaining the "steady state" optimal policy directly.

Consider a numerical example of the two state problem. A computer program was written to solve the general two state problem and is included in Appendix I.

The observer has constructed a 2 state model with parameters given and has the option of using a perfect channel or estimating the underlying Markov state and acting accordingly.

The probabilities describing the underlying process and the output channels are:

$$P = \begin{bmatrix} .9 & .1 \\ .4 & .6 \end{bmatrix} \qquad F = \begin{bmatrix} .9 & .1 \\ .3 & .7 \end{bmatrix}$$

The associated rewards are:

$$L = \begin{bmatrix} 3 & -2 \\ -6 & 4 \end{bmatrix}$$

A = cost of use of perfect information output channel

= 5 ~ loss of immediate earnings for one time unit

$$F^{o}[\underline{s(0)}] = 0.0 \qquad \text{for all } \underline{s(n)}$$

This could be the previous communications satellite example with the "perfect" information being obtained by using one unit of time to send a special test message to the satellite. The test message would be returned to the sender and from it he could glean the "perfect" information.

The computational results are show in figure 7. optimal policy is :

$0 \leq s_1(n) \leq .4$: estimate underlying Markov state 2 and act accordingly

$.4 \leq s_1(n) \leq 1.0$: estimate underlying Markov state 1 and act accordingly

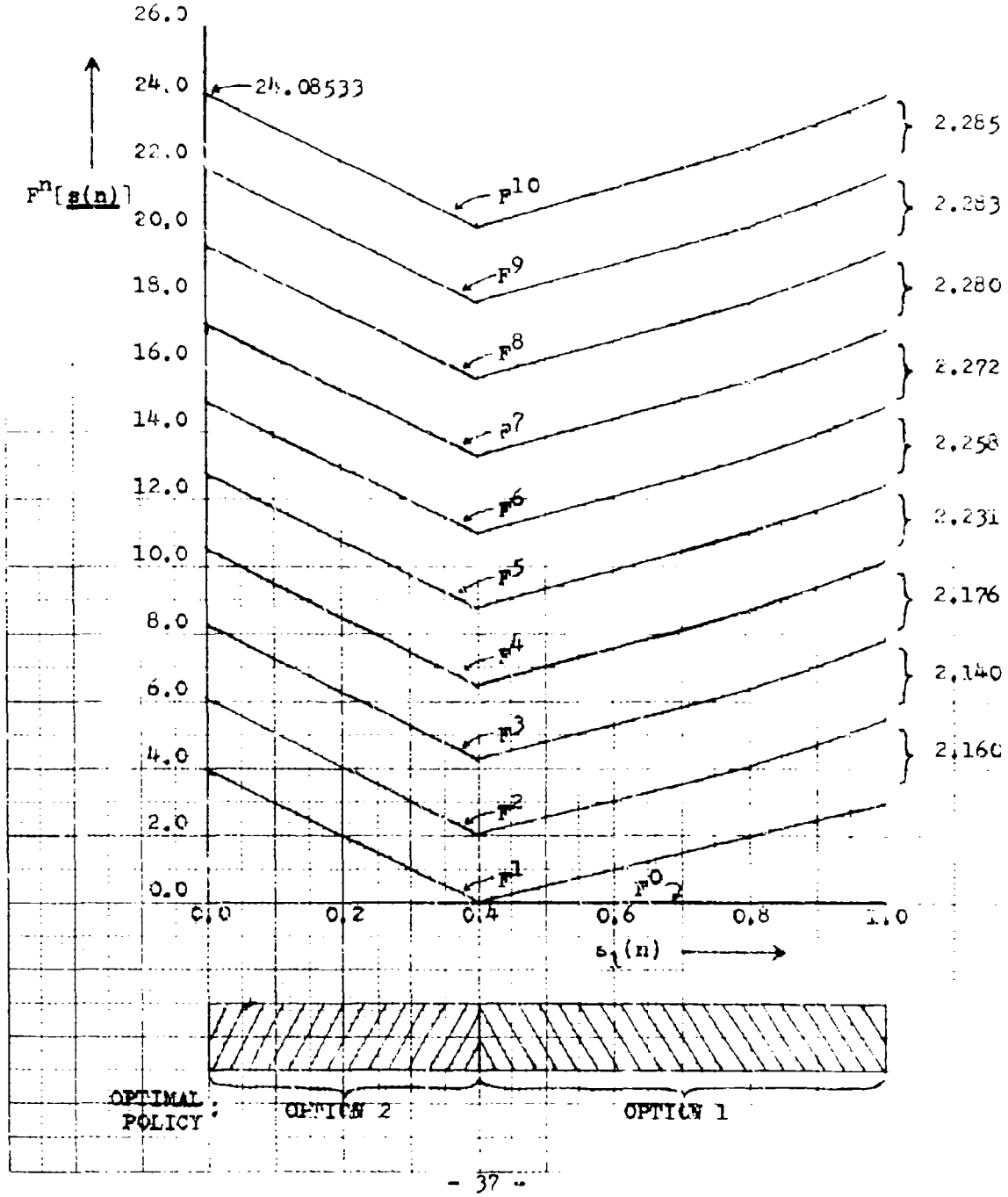
The perfect information channel is never used in the optimal policy. In relation to the other rewards, the cost of using the perfect information channel was too high. This problem is then equivalent to having only the options 1 and 2.

There is a growth pattern emerging as time (n) increases. In figure 7, the curves for $F^{n}[\underline{s(n)}]$ tend toward a fixed shape and the separation between $F^{n}[\underline{s(n)}]$ and $F^{n-1}[\underline{s(n-1)}]$ appears to be approaching 2.29 as n increases. The optimal policy is independent of time (n).

In the previous example the cost of perfect information

Figure 7

Total Expected Earnings
Using Optimal Policy



was too high to warrant usage of the "noiseless" channel. More illustrative results are obtained if the cost of perfect information is reduced to be:

A = loss of immediate earnings for one time unit.

The results of the calculations are given in figure 8. The optimal policy is:

n=1:

$0.0 \leq s_1(n) \leq 0.4$	estimate state 2 and act accordingly
$0.4 \leq s_1(n) \leq 1.0$	estimate state 1 and act accordingly

n=2:

$0.0 \leq s_1(n) \leq 0.4$	estimate state 2 and act accordingly
$0.4 \leq s_1(n) \leq 1.0$	estimate state 1 and act accordingly

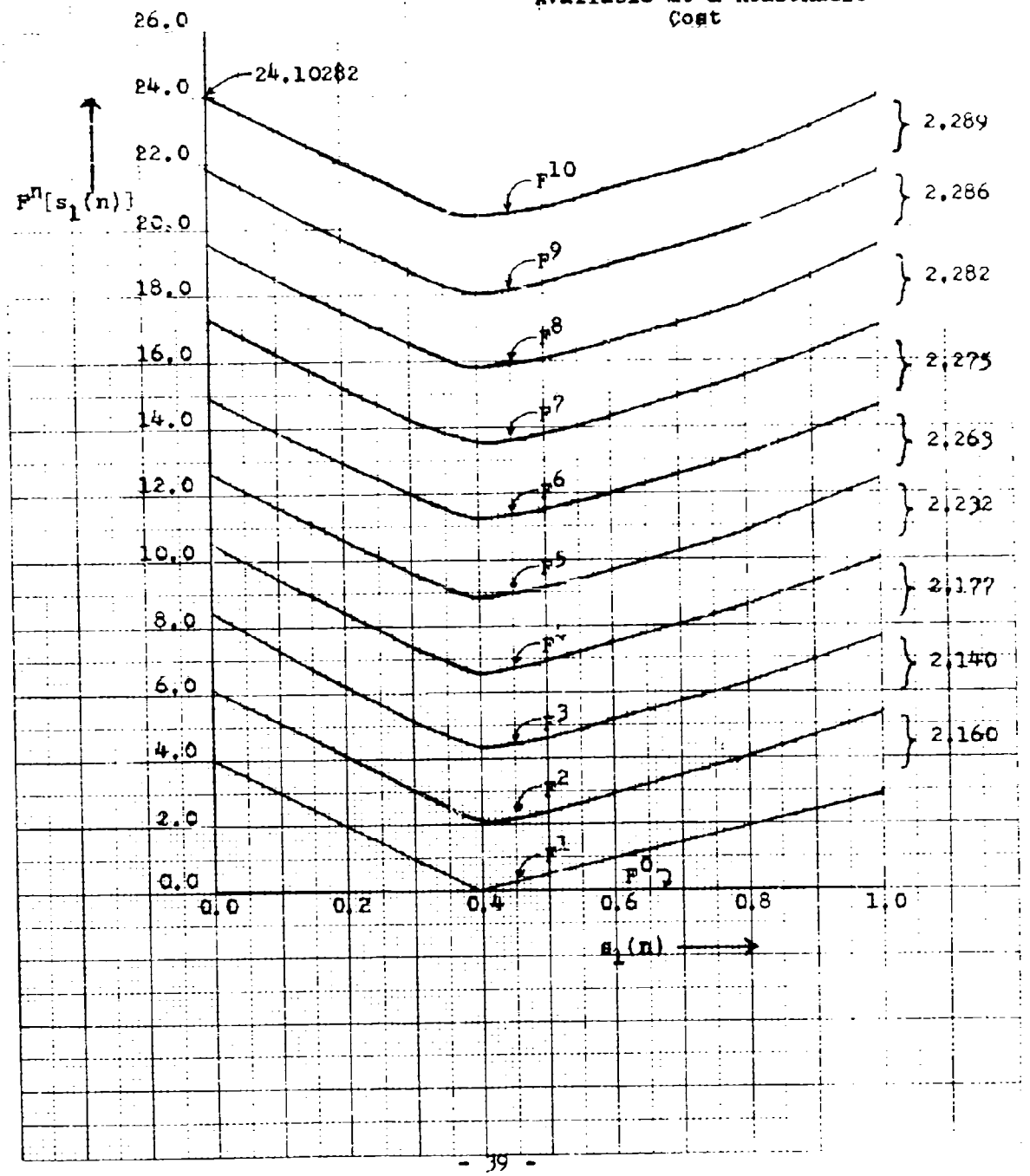
n=3,4,...9,10,

$0.0 \leq s_1(n) \leq .38$	estimate state 2 and act accordingly
$.38 < s_1(n) < .42$	use perfect information source
$.42 \leq s_1(n) \leq 1.0$	estimate state 1 and act accordingly

Here again a growth pattern on expected earnings is becoming discernible as n increases. The separation (gain)

Figure 8

Total Expected Earnings Using Optimal Policy with Perfect Information Available at a Reasonable Cost



between the $F^n[\underline{s(n)}]$ and $F^{n-1}[\underline{s(n-1)}]$ curves appears to be approaching 2.30 as n increases. Notice that when reasonably priced perfect information became available the growth of expected earnings as a function of time apparently increased slightly from 2.29 to 2.30.

2.9 Comments

This particular solution technique is useful in determining optimal policies associated with partially observable Markov processes for small time(n). The functional equations which must be solved for the optimal policies are of the general form given below where k(n) is the policy choice at time n.

$$F^n[\underline{s(n)}] = \max_k \left\{ q_k(\underline{s(n)}) + \sum_i P_i^k[\underline{s(n)}]F^{n-1}[\underline{T_i[s(n)]}] \right\}$$

While it is theoretically possible to obtain a solution to the above equation, it may be computationally infeasible. It is relatively easy and fast to obtain a numerical solution for the two state partially observable Markov process, but increasing the system to even five states may be prohibitive. A good deal of the strength of this technique depends on the analyst's ability to model the real world system with a few pertinent states.

This technique also becomes impractical when there is a large time (n) involved. The next chapter deals with the question of the existence of a steady state policy for a

partially observable Markovian system and methods of determining it without iteratively solving the functional equations introduced previously for ever increasing values of time.

CHAPTER III

OPTIMAL STEADY STATE POLICY DETERMINATION

In the chapter on optimal time dependent policies, as time grew large, the expected earnings seemed to converge on a discernible growth pattern and the policy was apparently becoming independent of the time (n) for large n. In many physical situations the time (n) which remains for the real world system to operate is large and sometimes even unknown. In those two situations it is not feasible to use the time dependent solution technique for optimal policy determination. The question of the existance of a steady state policy and a method of determining it becomes paramount.

The examples of Chapter II appeared to show the expected earnings, $F^n[\underline{s(n)}]$, converging on a growth pattern in the following manner for large time n.

$$F^n[\underline{s(n)}] = \max_k \left\{ q_k[\underline{s(n)}] + \sum_i P_i^k[\underline{s(n)}] F^{n-1}[T_i[\underline{s(n)}]] \right\}$$

$$F^n[\underline{s(n)}] \longrightarrow v[\underline{s(n)}] + nG \qquad \text{(large n)}$$

where $v[\underline{s(n)}]$ is interpreted as setting the steady state shape of the $F^n[\underline{s(n)}]$ curves and G, gain, is the steady state growth per unit time.

3.1 The State of Knowledge as a Continuous State Markov Process

To investigate the growth pattern of expected earnings,

it will be necessary to alter the concept of a partially observable Markov process model. Formerly it was interpreted as a descrete N state Markov process with stochastic output channels. Refocus now on the state of knowledge vector, $\underline{s(n)}$. It has n components, $s_i(n)$, which are constrained by:

$$0 \leq s_i(n) \leq 1 \qquad \sum_i s_i(n) = 1$$

The state of knowledge vector has n-1 independent components and may thus be represented as a point in an n-1 dimensional space. The state of knowledge vector for a partially observable Markov process model is in fact the state variable for a continuous state Markov process.

Consider a three state underlying process with $\underline{s(n)} = \underline{s_1(n), s_2(n), s_3(n)}$. Since $s_3(n) = 1-s_1(n)-s_2(n)$, then $s_1(n)$ and $s_2(n)$ describe the observer's state of knowledge.

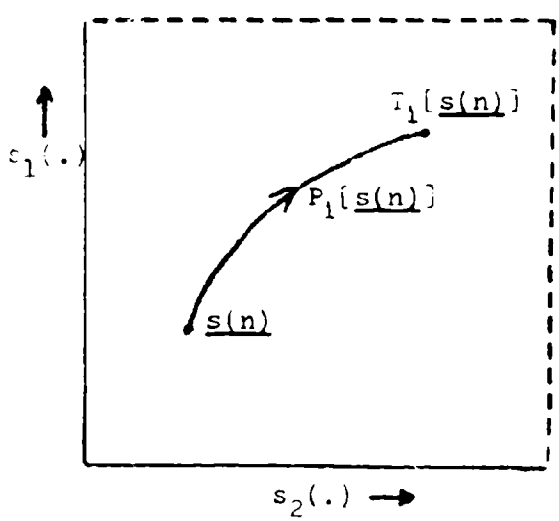


Figure 9 Continuous State Space

With probability $P_i[\underline{s(n)}]$, it will be transformation $T_i[\underline{s(n)}]$ which describes the new state in terms of the previous state, $\underline{s(n)}$. Since the transition probability depends only on the current state, the Markov assumption is satisfied. It happens that the state of knowledge is now also the state variable of a continuous state Markov process.

3.2 Steady State Gain

Let $h[\underline{s(n)}]$ be the probability density function on what the observer's state of knowledge will be at time n far into the future, given some initial state of knowledge. A completely ergodic Markov process is one whose limiting probability density function, $h[\underline{s(n)}]$, for n far into the future, is independent of the distribution of the starting state of knowledge.

The continuous state Markov process which has $\underline{s(n)}$ as its state variable cannot be considered to be completely ergodic. Suppose the initial state of knowledge vector for a two state partially observable process were:

$$\underline{s(n)} = \underline{c_o \ , \ 1 - c_o}$$

where c_o = a rational number

Note that the initial state of knowledge is precisely specified such that the initial density consists solely of an impulse at the point $s_1(n) = c_o$. As time progresses and the state of knowledge is repeatedly updated, $\underline{s(n)}$ will

always remain a vector whose components are rational numbers, by the nature of the transform applied. Therefore, the limiting probability density function on the state of knowledge will be nonzero only at a set of points selected from the rational numbers. Alternately, suppose that the initial distribution of the state of knowledge is described by a continuous density function. The limiting density $h[\underline{s(n)}]$ for such an initial state of knowledge distribution will be nonzero at points both inside and outside the set of rational numbers. Thus, the process is not completely ergodic.

Karlin[12] investigates the limiting steady state distribution in similar problems. A limiting density, $h[\underline{s(n)}]$, may exist for the class of initial densities which are continuous over some range of the allowable $\underline{s(n)}$. Drake[4] presents a method of computing the limiting density function for an arbitrary continuous initial density.

The steady state gain, G^k, for a given policy, $k(n)$, could be calculated using the following relation. Another method of determining the gain, G^k, is presented in the next section.

$$G^k = \int_{\underline{S(n)}} h^k[\underline{s(n)}]\, q_k[\underline{s(n)}]\, d\underline{s(n)}$$

3.3 Optimal Steady State Policy Determination

The existence of steady state gain implies that expected earnings, $F^n[\underline{s(n)}]$, converge on a growth pattern

for large n, and that a steady state optimal policy exists. The optimal steady state policy could be found by maximizing the gain G.

$$\max_k \; G^k \Longrightarrow \text{optimal policy, } k_{opt}$$

A method of policy optimization which is based on gain maximization was developed by Howard for use on decrete state Markov systems. His algorithm can be adapted for use in optimal policy determination on partially observable Markov process models.

Beginning with the basic equation for expected earnings:

$$F^n[\underline{s(n)}] = \max_k \left\{ q_k[\underline{s(n)}] + \sum_1 P_1^k[\underline{s(n)}] F^{n-1}[\underline{T_1[s(n)]}] \right\}$$

$$\longrightarrow \; v[\underline{s(n)}] + nG \qquad \text{for n large}$$

where G = steady state gain and $v[\underline{s(n)}]$ can be interpreted as initializing and setting the shape of the steady state expected earnings curve $F^n[\underline{s(n)}]$

Substituting the steady state form into the basic equation for a fixed (not necessarily optimal) policy, k:

$$v^k[\underline{s(n)}] + nG^k$$

$$= q_k[\underline{s(n)}] + \sum_1 P_1^k[\underline{s(n)}][v^k[\underline{T_1[s(n)]}] + (n-1)G^k]$$

$$= q_k[\underline{s(n)}] + (n-1)G^k + \sum_1 P_1^k[\underline{s(n)}] v^k[\underline{T_1[s(n)]}]$$

$$v^K[\underline{s(n)}] + G^K = q_K[\underline{s(n)}] + \sum_i P_i^K[\underline{s(n)}]v^K[\underline{T_i[s(n)]}]$$

$$\boxed{\begin{aligned} G^K &= q_K[\underline{s(n)}] - v^K[\underline{s(n)}] + \sum_i P_i^K[\underline{s(n)}]v^K[\underline{T_i[s(n)]}] \\ G^K &= \text{gain using a fixed policy } k \end{aligned}} \quad (*)$$

Solution of the above equation would yield the gain, G, and the curve $v[\underline{s(n)}]$ for a fixed policy. To solve the equation by computer, it is necessary to quantize the $v[\underline{s(n)}]$ curve into M points. Then these are a set of M simultaneous equations and M+1 unknowns. The M+1 unknowns are the M from $v[\underline{s(n)}]$ and 1 from G. Consider adding a quantity, c, to each point of the $v[\underline{s(n)}]$ curve.

$$G = q[\underline{s(n)}] - [v[\underline{s(n)}] + c] + \sum_i P_i[\underline{s(n)}][v[\underline{T_i[s(n)]}] + c]$$

$$= q[\underline{s(n)}] - v[\underline{s(n)}] - c + \sum_i P_i[\underline{s(n)}]c + \sum_i P_i[\underline{s(n)}]v[\underline{T_i[s(n)]}]$$

$$\text{but } \sum_i P_i[\underline{s(n)}]c = c \sum_i P_i[\underline{s(n)}] = c$$

$$G = q[\underline{s(n)}] - v[\underline{s(n)}] + \sum_i P_i[\underline{s(n)}]\, v[\underline{T_i[s(n)]}]$$

Notice that this is the same equation (*) back again. This implies that the absolute level of the $v[\underline{s(n)}]$ curve cannot be determined from these equations. Therefore to allow solution, arbitrarily fix one point on the curve.

$$v[\underline{1,0,\ldots,0}] = 0$$

As will be shown later, only the relative value of the $v[\underline{s(n)}]$ curve (i.e. the shape) is necessary. The M equations and M unknowns may be solved for the gain, G, and $v[\underline{s(n)}]$ subject to the above stipulation.

Howard's algorithm contains a policy improvement routine which rapidly converges on the optimal policy and thus it is necessary to solve the set of M equations for only a few policies. Later in this chapter, a method for checking the "optimal" solution is introduced so that errors introduced in solving the many simultaneous equations can be detected and corrected. Also, a technique for avoiding the need to solve the M simultaneous equations was developed by Schweitzer and will be introduced.

For a fixed policy k, the following is solved for G^k and $v^k[\underline{s(n)}]$.

$$G^k = q_k[\underline{s(n)}] - v^k[\underline{s(n)}] + \sum_i P_i{}^k[\underline{s(n)}]v^k[\underline{T_i[\underline{s(n)}]}]$$

The next step is to use the G^k and $v^k[\underline{s(n)}]$ obtained to find a better policy. Recall that the optimum policy maximized:

$$F^n[\underline{s(n)}] = \max_{k(n)} \left\{ q_{k(n)}[\underline{s(n)}] + \sum_i P_i{}^{k(n)}[\underline{s(n)}]F^{n-1}[\underline{T_i[\underline{s(n)}]}] \right\}$$

In the steady state:

$$v_{opt}[\underline{s(n)}] + n G_{opt} =$$

$$= \max_k \left\{ q_k[\underline{s(n)}] + \sum_i P_i^k[\underline{s(n)}][v_{opt}[\underline{T_i[s(n)]}] + (n-1)G_{opt}] \right\}$$

$$= \max_k \left\{ q_k[\underline{s(n)}] + (n-1)G_{opt} + \sum_i P_i^k[\underline{s(n)}] v_{opt}[\underline{T_i[s(n)]}] \right\}$$

Since G_{opt} here is the gain associated with the optimal policy, it is not a function of k and an equvalent test quantity to be maximized as a function of policy (k) is.

$$\text{TEST}[\underline{s(n)}] = q_k[\underline{s(n)}] + \sum_i P_i^k[\underline{s(n)}] v_{opt}[\underline{T_i[s(n)]}]$$

Because $\sum_i P_i^k[\underline{s(n)}] = 1$, any additive constant in $v_{opt}[\underline{T_i[s(n)]}]$ would not affect the test quantity and therefore only relative values of $v_{opt}[\underline{T_i[s(n)]}]$ are needed.

Howard's algorithm says that maximizing the test quantity using $v[\underline{s(n)}]$ from some arbitrary policy (not necessarily optimal) will always yield a new policy which is at least as good as the old arbitrary one and that an optimal policy can be found in the manner illustrated in figure 10.

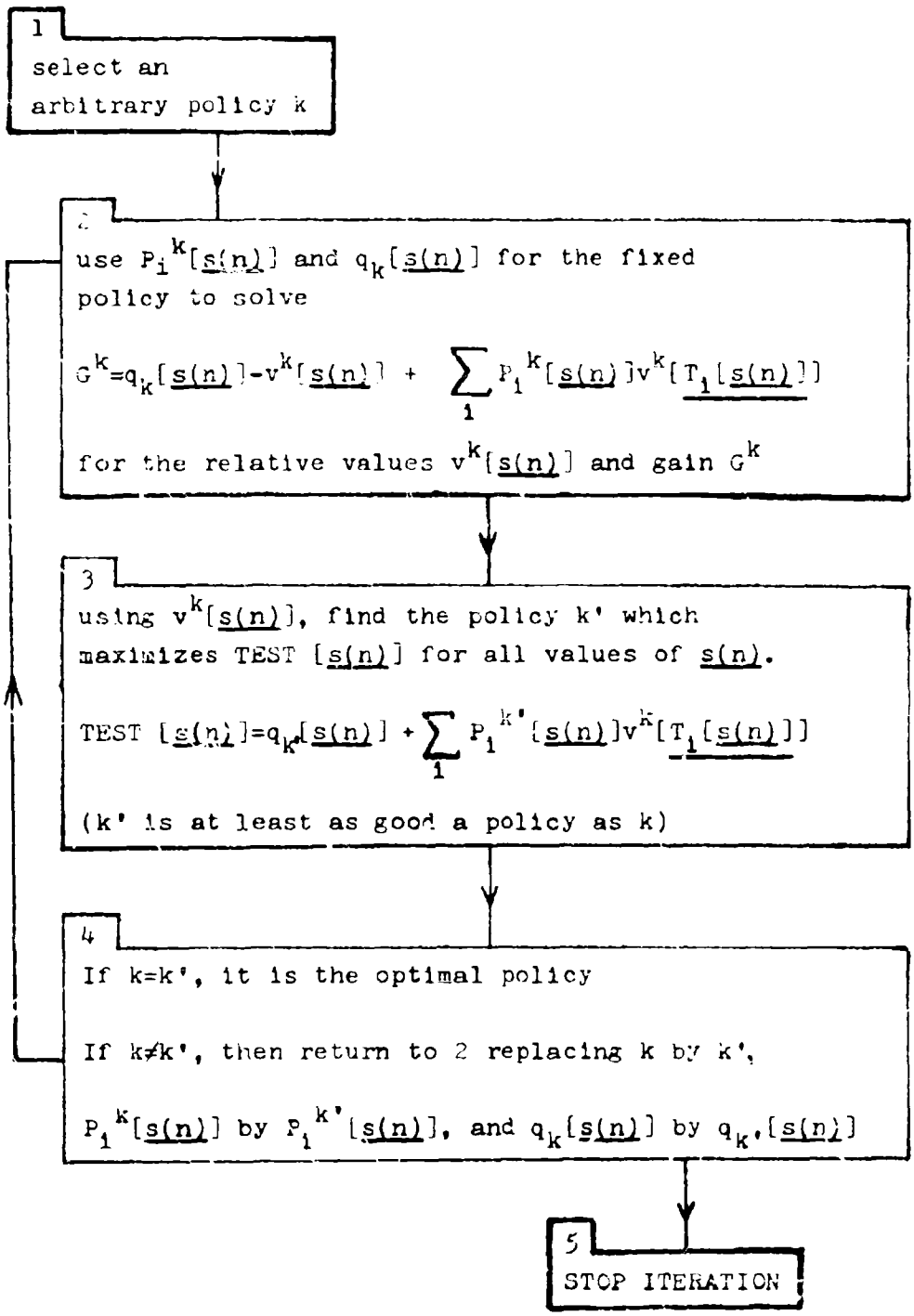


Figure 10 Determination of the Optimal Steady State Policy

By refocusing attention from the interpretation of a partially observable Markov process as an underlying descrete Markov process plus output channels, it is seen that the state of knowledge vector is the state variable of a continuous state Markov process. The optimal policy can be found utilizing a digital computer and Howard's algorithm for descrete Markov processes. How close the digital solution is to the continuous state solution remains to be seen.

Experience with descrete state problems has shown Howard's algorithm to be computationally efficient and that the sequence of policies generated iteratively will usually converge in a small number of cycles. The convergence may be hastened by selecting the arbitrary initial policy to be as close to optimal as possible. The decision-maker could incorporate all of his prior feelings into the initial policy although it is not necessary to do so.

3.4 Proof of Policy Convergence and Optimization

A proof, adapted from Howard's work, is now offered showing that the policy which is converged upon is in fact the one with the highest gain of all possible policies.

Suppose that an initial policy A has been operated upon and the policy improvement routine has produced a policy B which is different from A.

Prove : $G^B \geq G^A$

since B was chosen over A:

$$TEST^B[\underline{s(n)}] \geq TEST^A[\underline{s(n)}] \qquad \text{for all } \underline{s(n)}$$

$$q_B[\underline{s(n)}] + \sum_1 P_1^B[\underline{s(n)}]\, v^A[T_1[\underline{s(n)}]]$$

$$\geq q_A[\underline{s(n)}] + \sum_1 P_1^A[\underline{s(n)}] v^A[T_1[\underline{s(n)}]]$$

Let:

$$\gamma_A^B[\underline{s(n)}] = TEST^B[\underline{s(n)}] - TEST^A[\underline{s(n)}]$$

$$\gamma_A^B[\underline{s(n)}] = q_B[\underline{s(n)}] - q_A[\underline{s(n)}] + \sum_1 P_1^B[\underline{s(n)}] v^A[T_1[\underline{s(n)}]]$$

$$- \sum_1 P_1^A[\underline{s(n)}] v^A[T_1[\underline{s(n)}]]$$

$\gamma_A^B[\underline{s(n)}] \geq 0$ for all $\underline{s(n)}$

where $\gamma_A^B[\underline{s(n)}]$ is the improvement in the test quantity that the policy improvement routine was able to make.

The expressions for G^B and G^A are:

$$G^B = q_B[\underline{s(n)}] - v^B[\underline{s(n)}] + \sum_1 P_1^B[\underline{s(n)}] v^B[T_1[\underline{s(n)}]]$$

$$G^A = q_A[\underline{s(n)}] - v^A[\underline{s(n)}] + \sum_1 P_1^A[\underline{s(n)}] v^A[T_1[\underline{s(n)}]]$$

Subtracting G_A from G_B and rearranging:

$$G^B - G^A + v^B[\underline{s(n)}] - v^A[\underline{s(n)}] = q_B[\underline{s(n)}] - q_A[\underline{s(n)}]$$

$$+ \sum_1 P_1^B[\underline{s(n)}] v^B[T_1[\underline{s(n)}]]$$

$$- \sum_1 P_1^A[\underline{s(n)}] v^A[T_1[\underline{s(n)}]]$$

Introducing $\gamma_A^B(n)$:

$$G^B - G^A + v^B[\underline{s(n)}] - v^A[\underline{s(n)}] = \gamma_B^A[\underline{s(n)}] + \sum_i P_i^B[\underline{s(n)}]v^B[\underline{T_i[s(n)]}] - \sum_i P_i^B[\underline{s(n)}]v^A[\underline{T_i[s(n)]}]$$

$$G^B - G^A + v^B[\underline{s(n)}] - v^A[\underline{s(n)}] = \gamma_A^B[\underline{s(n)}] + \sum_i P_i^B[\underline{s(n)}]\{v^B[\underline{T_i[s(n)]}] - v^A[\underline{T_i[s(n)]}]\}$$

Define:

$$\Delta G = G^B - G^A$$

$$\Delta v[\underline{s(n)}] = v^B[\underline{s(n)}] - v^A[\underline{s(n)}]$$

Substituting the definitions into the previous equation:

$$\Delta G + \Delta v[\underline{s(n)}] = \gamma_A^B[\underline{s(n)}] + \sum_i P_i^B[\underline{s(n)}]\Delta v[\underline{T_i[s(n)]}]$$

1) $$\Delta G = \gamma_A^B[\underline{s(n)}] - \Delta v[\underline{s(n)}] + \sum_i P_i^B[\underline{s(n)}]\Delta v[\underline{T_i[s(n)]}]$$

Note that the above is identical in form to:

2) $$T = q[\underline{s(n)}] - v[\underline{s(n)}] + \sum_i P_i[\underline{s(n)}]v[\underline{T_i[s(n)]}]$$

Recall that the steady state probability density, $h[\underline{s(n)}]$, was related to G in the following manner:

$$3) \quad G = \int_{\underline{S(n)}} [h[\underline{s(n)}]q[\underline{s(n)}]]d\underline{s(n)}$$

So the solution for ΔG in equation . is:

$$4) \quad \Delta G = \int_{\underline{S(n)}} [h^B[\underline{s(n)}]\ \gamma_A^B[\underline{s(n)}]]\ d\underline{s(n)}$$

Since:

$$\gamma_A^B[\underline{s(n)}] \geq 0 \qquad \text{for all } \underline{s(n)}$$

and a property of all probability density functions is:

$$h^B[\underline{s(n)}] \geq 0 \qquad \text{for all } \underline{s(n)}$$

Therefore:

$$\Delta G \geq 0$$

A new policy obtained using the algorithm has at least as high a steady state gain as the old policy. Furthermore, it is impossible for a policy with higher steady state gain to exist and not be discovered ultimately by the iterative routine.

Assume for two policies X and Y, that $G^Y > G^X$ but that the iteration routine has converged on X.

Since X was chosen over Y in the policy improvement routine and the policy X set the test policy just prior to convergence:

$$TEST^X[\underline{s(n)}] \geq TEST^Y[\underline{s(n)}]$$

$$q_X[\underline{s(n)}] + \sum_1 P_1^X[\underline{s(n)}]v^X[\underline{s(n)}] \geq q_Y[\underline{s(n)}] + \sum_1 P_1^Y[\underline{s(n)}]v^X[\underline{s(n)}]$$

$$\delta_Y^X[\underline{s(n)}] = TEST^X[\underline{s(n)}] - TEST^Y[\underline{s(n)}] \geq 0$$

By the method of the previous proof:

$$\Delta G = G^X - G^Y \geq 0$$

Since the initial assumption was that $G^Y > G^X$, this is a direct contradiction and hence it is impossible for the algorithm to ultimately converge on a policy which has less than optimal gain.

3.5 Reinterpretation of Relative Values

It is not immediately obvious that the relative values, $v[\underline{s(n)}]$, which are obtained for one fixed policy, should be useful in policy improvement. To obtain some insight into

this matter and into the general concept of steady state policy determination, consider a "policy space" which consists of all conceivable policies. For any policy, k, the state of knowledge vector, $\underline{s(n)}$, can undergo certain transformations, $T_i[\underline{s(n)}]$, to obtain a new "state of knowledge" vector. The variables that are a function of policy k, are the immediate earnings, $q_k[\underline{s(n)}]$, and the probabilities that specific transformations are applied, $P_i^k[\underline{s(n)}]$. If we have M possible transformations, there are M independent functions in general that are set by the policy. There are M-1 independent $P_i^k[\underline{s(n)}]$ and one $q_k[\underline{s(n)}]$. A policy fixes the decision to be made for each $\underline{s(n)}$. Consider one specific but arbitrary $\underline{s(n)}$. For this value of $\underline{s(n)}$, the M independent parameters associated with the decision would specify a point in a Euclidean M-space. Thus, every conceivable decision could be represented by a point in this "decision space". Only certain of those points would represent allowable decisions.

Consider two decisions, A and B, with points D_A and D_B in the decision space for a particular $\underline{s(n)}$. Now consider all possible decisions lying on the line segment joining D_A and D_B in the decision space. Pick a new decision on that line segment and define it to be a "randomization" of A and B. If r is the randomization parameter and AB is the randomized decision, the new variables are related to the A and B variables thusly.

$$P_1^{AB}[r,\underline{s(n)}] = rP_1^{B}[\underline{s(n)}] + (1-r)P_1^{A}[\underline{s(n)}]$$

$$q_{AB}[r,\underline{s(n)}] = rq_B[\underline{s(n)}] + (1-r)q_A[\underline{s(n)}]$$

The gains of policies A and B are:

$$G^A = \int h^A[\underline{s(n)}]q_A[\underline{s(n)}]d\underline{s(n)}$$

$$G^B = \int h^B[\underline{s(n)}]q_B[\underline{s(n)}]d\underline{s(n)}$$

The gain of the randomized policy is:

$$G^{AB}(r) = \int [h^{AB}[r,\underline{s(n)}]q_{AB}[r,\underline{s(n)}]]d\underline{s(n)}$$

where

$$h^{AB}[r,\underline{s(n)}] = rh^B[\underline{s(n)}] + (1-r)h^A[\underline{s(n)}]$$

Relating this back to policy improvement, consider policy A as the initial policy and policy AB as the policy being tested.

$$\delta_A^{AB}[r,\underline{s(n)}] = \text{improvement in test quantity}$$

$$= q_{AB}[r,\underline{s(n)}] - q_A[\underline{s(n)}]$$

$$+ \sum_i P_1^{AB}[r,\underline{s(n)}]v^A[\underline{T_i[s(n)]}]$$

$$- \sum_i P_1^{A}[\underline{s(n)}]v^A[\underline{T_i[s(n)]}]$$

$$\delta_A^{AB}[r,\underline{s(n)}] = rq_B[\underline{s(n)}] + (1-r)\, q_A[\underline{s(n)}] - q_A[\underline{s(n)}]$$

$$+ \sum_i (rP_i^B[\underline{s(n)}] + (1-r)\, P_i^A[\underline{s(n)}])\, v^A[\underline{T_i[\underline{s(n)}]}]$$

$$- \sum_i P_i^A[\underline{s(n)}]\, v^A[\underline{T_i[\underline{s(n)}]}]$$

$$= r\left\{ (q_B[\underline{s(n)}] - q_A[\underline{s(n)}]) + \sum_i P_i^B[\underline{s(n)}]\, v^A[\underline{T_i[\underline{s(n)}]}] \right.$$

$$\left. - \sum_i P_i^A[\underline{s(n)}]\, v^A[\underline{T_i[\underline{s(n)}]}] \right\}$$

$$\delta_A^{AB}[r,\underline{s(n)}] = r\, \delta_A^B[\underline{s(n)}]$$

$$G^{AB} - G^A = \int h^{AB}[r,\underline{s(n)}]\, \delta_A^{AB}[r,\underline{s(n)}]\, d\underline{s(n)}$$

$$G^{AB} - G^A = r \int h^{AB}[r,\underline{s(n)}]\, \delta_A^B[\underline{s(n)}]\, d\underline{s(n)}$$

Now let the randomized decision approach the original decision. Divide by r and take the limit as $r \to 0$.

$$\lim_{r\to 0} \frac{G^{AB}(r) - G^A}{r} = \frac{\partial G^{AB}(r)}{\partial r}\Bigg]_{r=0} = \int h^A[\underline{s(n)}]\, \delta_A^B[\underline{s(n)}]\, d\underline{s(n)}$$

Since $\delta_A^B[\underline{s(n)}]$ is the improvement in the test quantity of the policy improvement routine if decision B is substituted for original decision A, and since $h^A[\underline{s(n)}] \geq 0$ for all $\underline{s(n)}$, then finding the decision B which gives maximum

test quantity improvement is equivalent to finding:

$$\max_{B} \left[\frac{\partial G^{AB}(r)}{\partial r} \right]_{r=0}$$

For a given policy A, the algorithm computes (for each B) the directional derivative, evaluated at A, of gain from A to B in decision space. It then selects as the new policy the one with the highest directional derivative of gain. If A is the optimal policy then all the directional derivatives evaluated at A will be less than or equal to zero. With this interpretation of what the policy improvement routine does, it becomes clearer why the relative values, $v[\underline{s(n)}]$ at the old policy A are useful. They determine $\delta_A^B[\underline{s(n)}]$ which in turn is closely related to the directional derivative of gain evaluated at A.

3.6 Example

Using the technique just developed for steady state solution for the optimal policy, consider the numerical example introduced in Chapter II.

$$[P] = \begin{bmatrix} .9 & .1 \\ .4 & .6 \end{bmatrix} \qquad [F] = \begin{bmatrix} .9 & .1 \\ .3 & .7 \end{bmatrix}$$

A = cost of use of the perfect information channel.

A = 5 + loss of immediate earnings for one time unit.

$$F^{0}[\underline{s(0)}] = 0 \qquad \text{for all } \underline{s(0)}$$

A computer program has been written to find the steady state gain, relative values, and optimal policy for a two state partially observable Markov process and is included in Appendix II.

In this example the decision-maker has three options open to him.

option 1 : continue transmitting until the next output is received

option 2 : stop transmitting and check again one time unit later

option 3 : use perfect information channel

Recall that in Chapter II it was found that the cost of perfect information was too high to warrant use of the "noiseless" channel. This was found also to be true for the steady state policy here. The results are given in Table I and Figure 11 provides a graphical comparison of the steady state relative values and the $F^{10}[\underline{s(10)}]$ expected earnings curve found in Chapter II. They are almost identical in shape. By looking at the transient expected earnings, the gain was predicted to be about 2.29 and the steady state gain was found to be 2.29897 using the computer program of Appendix II and the techniques introduced in this chapter. With the cost of perfect information reduced to a reasonable level the example was reworked.

TABLE I

EXAMPLE RESULTS

With no perfect information channel available the gain in steady state was:

$$G = 2.29897$$

The relative values and decisions are:

$s_1(n)$	$v(\underline{s(n)})$	Decision	$s_1(n)$	$v(\underline{s(n)})$	Decision
0.00	+0.21580	2	.52	-3.02341	1
.02	-0.01476	2	.54	-2.91813	1
.04	-0.22287	2	.56	-2.92431	1
.06	-0.39100	2	.58	-2.69816	1
.08	-0.59868	2	.60	-2.64197	1
.10	-0.77769	2	.62	-2.53330	1
.12	-0.98507	2	.64	-2.42463	1
.14	-1.27914	2	.66	-2.24044	1
.16	-1.41860	2	.68	-2.12952	1
.18	-1.62406	2	.70	-2.05787	1
.20	-1.72908	2	.72	-2.00757	1
.22	-1.93353	2	.74	-1.89419	1
.24	-2.14215	2	.76	-1.72680	1
.26	-2.38346	2	.78	-1.61142	1
.28	-2.58545	2	.80	-1.54021	1
.30	-2.74258	2	.82	-1.47230	1
.32	-2.99491	2	.84	-1.19940	1
.34	-3.19564	2	.86	-1.05639	1
.36	-3.39637	2	.88	-0.93836	1
.38	-3.62377	2	.90	-0.76434	1
.40	-3.82286	1 or 2	.92	-0.63339	1
.42	-3.72195	1	.94	-0.38545	1
.44	-3.43766	1	.96	-0.24961	1
.46	-3.36893	1	.98	-0.09156	1
.48	-3.26575	1	1.00	-0.00000	1
.50	-3.16257	1			

TABLE II

EXAMPLE RESULTS

With perfect information available the gain in steady state was:

$$G = 2.29958$$

The relative values and decisions are:

$s_1(n)$	$v(\underline{s(n)})$	Decision	$s_1(n)$	$v(\underline{s(n)})$	Decision
0.00	+0.18809	2	.52	-3.05064	1
.02	-0.04244	2	.54	-2.94554	1
.04	-0.25055	2	.56	-2.83334	1
.06	-0.41867	2	.58	-2.74047	1
.08	-0.62634	2	.60	-2.66937	1
.10	-0.80534	2	.62	-2.56069	1
.12	-1.01271	2	.64	-2.45301	1
.14	-1.30676	2	.66	-2.26706	1
.16	-1.44621	2	.68	-2.15612	1
.18	-1.65166	2	.70	-2.08446	1
.20	-1.75667	2	.72	-2.03414	1
.22	-1.96112	2	.74	-1.93074	1
.24	-2.27965	2	.76	-1.75335	1
.26	-2.41102	2	.78	-1.63796	1
.28	-2.61301	2	.80	-1.56673	1
.30	-2.77013	2	.82	-1.49880	1
.32	-3.02245	2	.84	-1.21792	1
.34	-3.22317	2	.86	-1.06489	1
.36	-3.42389	2	.88	-0.90460	1
.38	-3.65127	2	.90	-0.78979	1
.40	-3.81191	3	.92	-0.65884	1
.42	-3.74944	1	.94	-0.41093	1
.44	-3.45977	1	.96	-0.27510	1
.46	-3.39098	1	.98	-0.11798	1
.48	-3.28774	1	1.00	-0.00000	1
.50	-3.19451	1			

Figure 11
Steady State v[s(n)]
without Perfect
Information

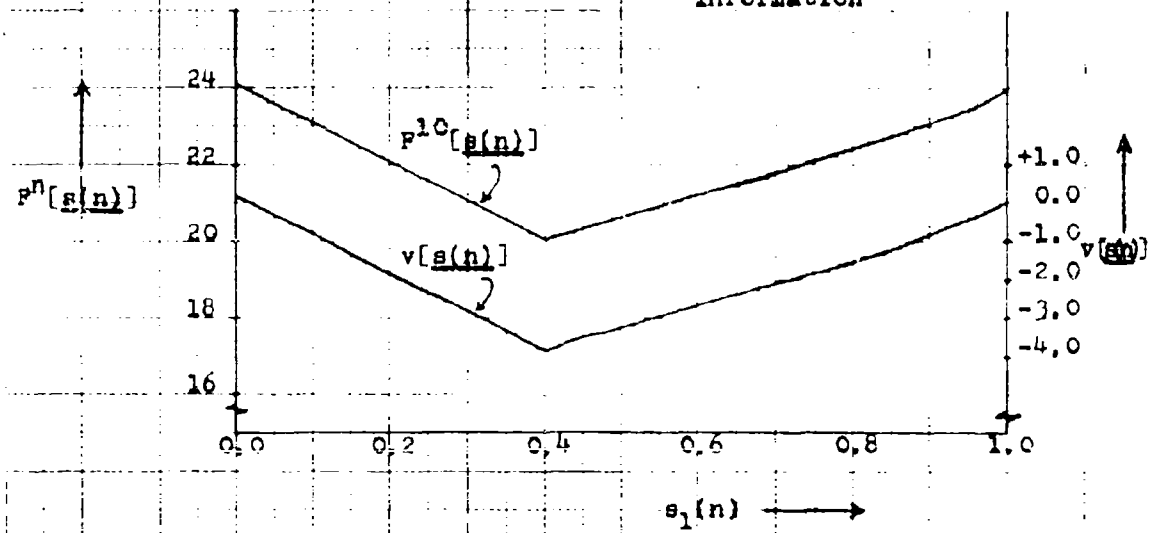
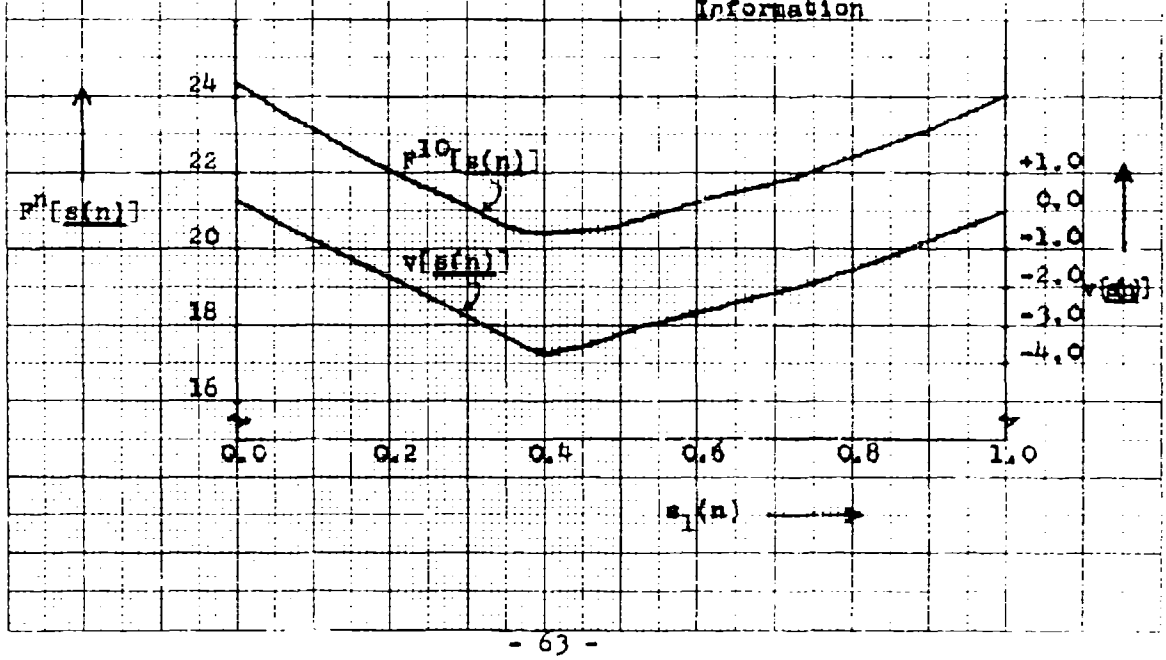


Figure 12
Steady State v[s(n)]
With Perfect
Information



A = cost of use of perfect information channel.
= loss of immediate earnings for one time unit.

The steady state gain, relative values, and policy are given in Table II and Figure 1. provides a graphical comparison of steady state relative values and $F^{10}[\underline{s(10)}]$, the expected earnings found when the same example was worked in Chapter II. Again the curves are nearly identical in shape and the steady state gain of 2.9998 is near the 3.30 previously predicted. The results obtained by the two different techniques support each other. Notice that the availability of perfect information increased the gain.

If finer precision is desired in the range of $\underline{s(n)}$ where a change of "best option" takes place, a finer grid could be used in that region. That is, in breaking the continuous $\underline{s(n)}$ vector into descrete points, make more divisions in regions of particular interest.

3.7 Verifying the Numerically Determined Policy

Because the computer solution involves a descrete approximation to the continuous $\underline{s(n)}$ vector, the numerically produced "optimal" solution could vary from the true optimal solution. To check on the accuracy of the numerical solution, one might vary the number of descrete points used to approximate the continuous vector and note the effect this has on the solution. A much better technique is to test the steady state solution in question by use of the time-dependent techniques of Chapter II. Recall that the observer specifies some

initial expected earnings, $F^0[\underline{s(0)}]$, and the time-dependent techniques of Chapter II compute $F^1[\underline{s(1)}]$ and so forth iteratively. The steady state relative values were interpreted as specifying the shape of the $F^n[\underline{s(n)}]$ expected earnings curve for n large. Therefore if the true steady state relative values, $v[\underline{s(n)}]$, were used as the initial expected earnings $F^0[\underline{s(0)}]$, then $F^1[\underline{s(1)}]$ should be simply $F^0[\underline{s(0)}]$ plus the steady state gain.

$$F^1[\underline{s(1)}] = F^0[\underline{s(0)}] + G = v[\underline{s(n)}] + G$$

This provides a check on the numerical steady state policy which may be in question.

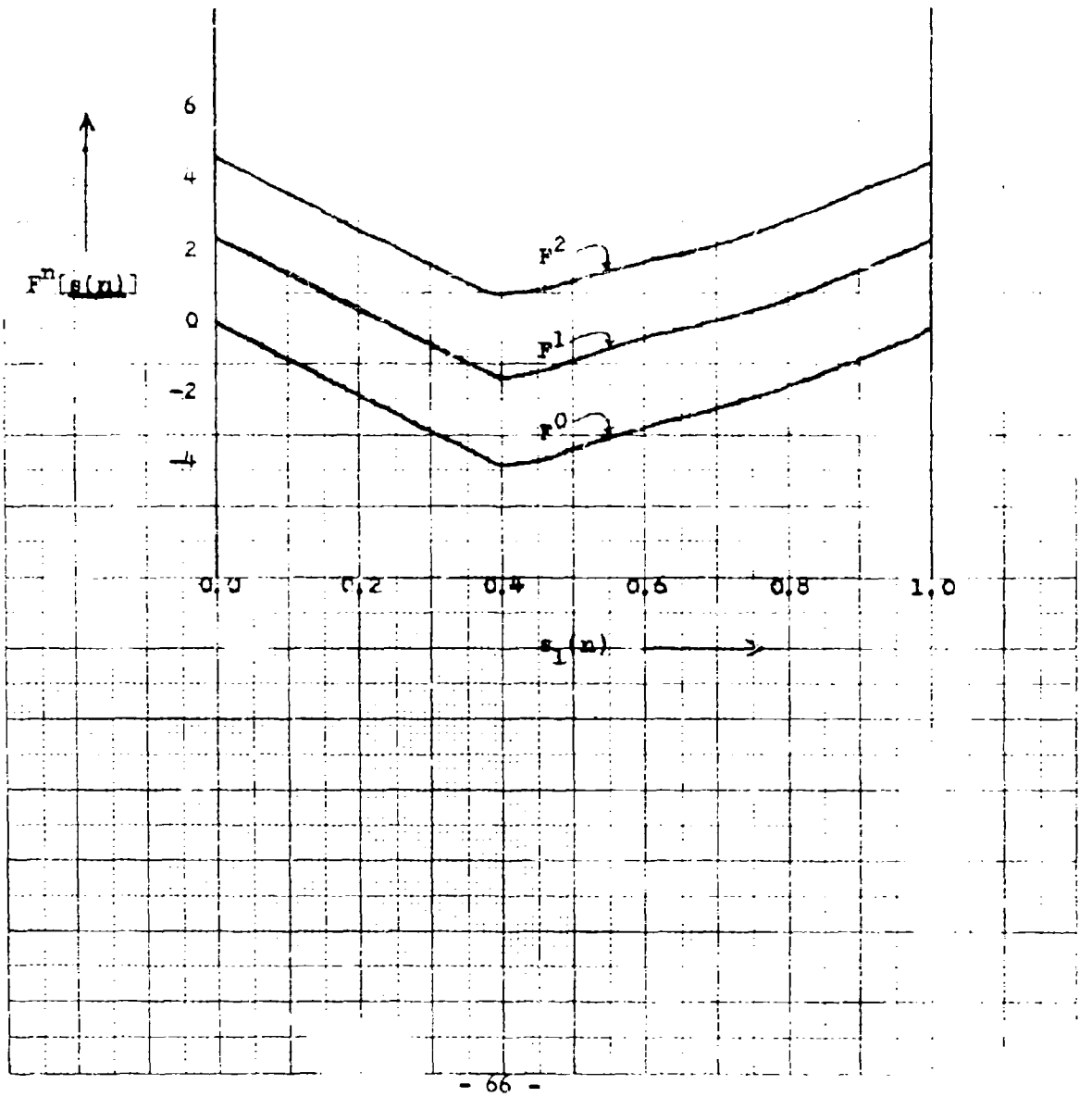
Figure 13 shows the result of such a check which was run on the example which has been used throughout this report.

3.8 Computational Considerations

Schweitzer[10] developed an improvement on Howard's algorithm for descrete state Markov processes which makes it computationally more practical for problems with a large number of states. Normally the policy improvement portion of the routine is used on all of the states before solving the M simultaneous equations of the value determination portion of the cycle. Schweitzer noted that if the policy improvement was done on only one state, the solution to the new set of simultaneous equations is quite simply

Figure 13

Testing the Steady State Solution



related to the solution of the old set of simultaneous equations. By judicious choice of the arbitrary initial policy, the initial solution to the simultaneous equations is trivial. Effectively, the set of equations need never be solved and a major deterent to the use of the algorithm has been removed. Schweitzer estimates that with his modification, Howard's algorithm would be able to handle on the order of five thousand discrete states.

Thus, dynamic programming techniques have been shown to be of use in the determination of optimal steady state policies associated with partially observable Markov processes.

CHAPTER IV

CONCLUDING REMARKS

The partially observable Markov process has been presented and some of its properties discussed. The primary area of investigation in this report was the selection of a course of action from a set of alternatives using only the information about the system which is available from the observable outputs.

Dynamic programming techniques were shown to be of use in the optimization of both transient and steady state policies. While theoretically the optimization can always be done, there is a definite computational limitation which was discussed.

There are several extensions of this investigation that could be made. The concept of discounting of future rewards could be considered. The optimum placement of investment dollars to improve prediction abilities and average earnings could be investigated. In addition, time-variant system parameters could be introduced.

The optimization technique used deals with a problem of much higher dimensionality than that of the original underlying process. A method is needed which will allow solution of sequential decision problems beyond the scope of those that can be handled by the technique presented in this report.

APPENDIX I

COMPUTER SOLUTION FOR THE OPTIMAL TIME DEPENDENT POLICY

In chapter two, equations for a two state process are formulated. They are solved by this program for the case where the decision-maker has available the options of:

1) estimate state 1 and act accordingly.
2) estimate state 2 and act accordingly.
3) use perfect information channel at added cost.

For the two state process the state of knowledge vector $\underline{s_1(n), s_2(n)}$ is fully specified by $s_1(n)$ since $s_2(n) = 1-s_1(n)$. The program breaks $s_1(n)$ (which can take any value from 0.0 to 1.0) into 51 points and calculates the maximum expected utility of rewards, $F^n(\underline{s(n)})$, for each of the 51 points. The initial values $F^o(\underline{s(0)})$ must be specified and then the program calculates $F^1(\underline{s(1)})$ by finding the maximum of the three possible options rewards. $F^2(\underline{s(2)})$ is then found from $F^1(\underline{s(1)})$. This continues on up to $n=n_{max}$, which is specified by the person using the program. The output is in the form of expected total utility of rewards at each of the 51 points for each n from 1 to n_{max} plus the decision D(n,k) to be made at each point k for each time n. This decision is optimal basedon the criterion of maximizing the expected utility of rewards.

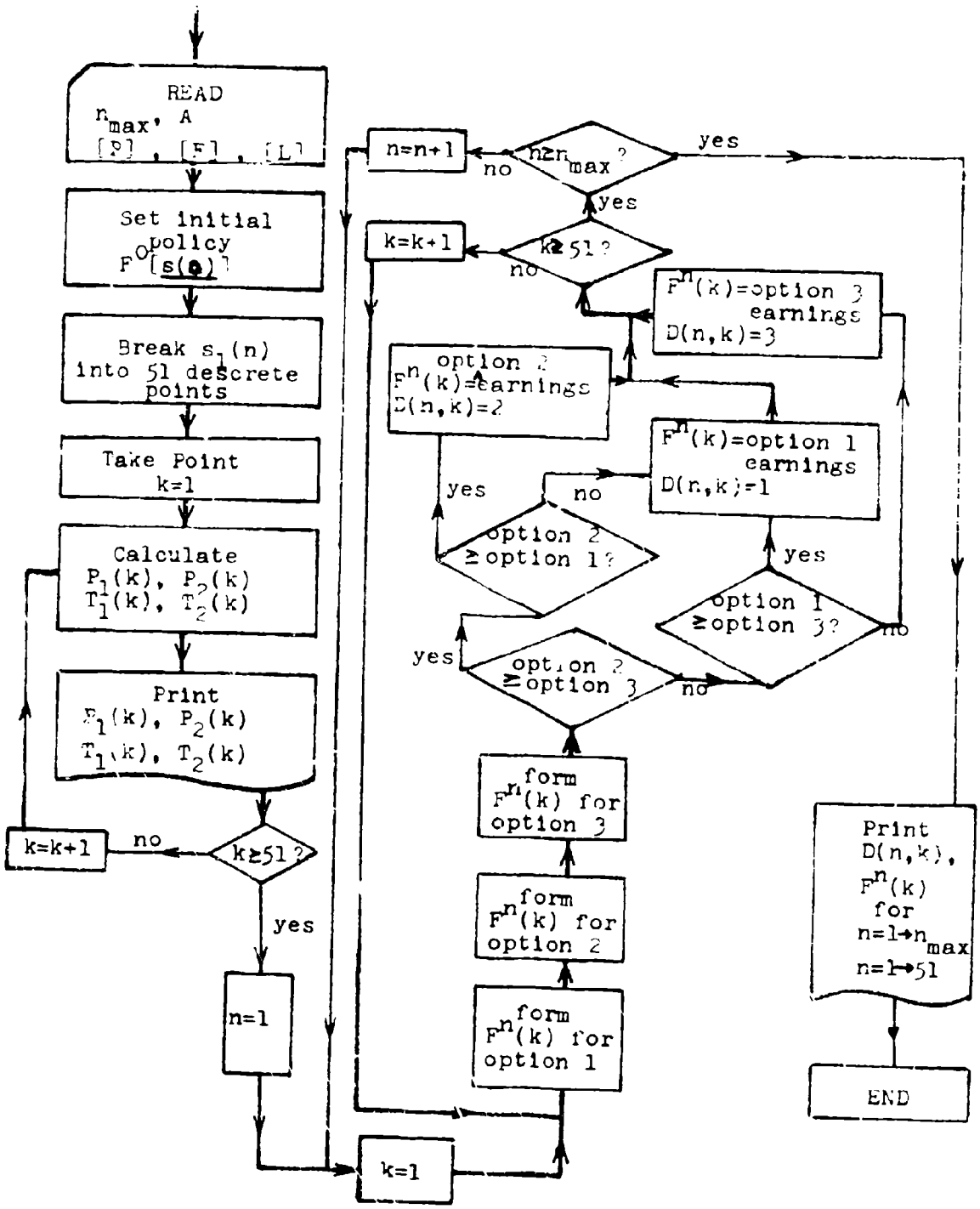


Figure 14 Flow Graph - "Value"

PROGRAM LISTING - "VALUE"

```
      DIMENSION E(11,51),P(2,2),F(2,2),R(2,2),X(51)
      DIMENSION PONE(51),PTWO(51),TONE(51),TTWO(51),LONE(51),LTWO(51)
      COMMON E,P,F,R,X,PONE,PTWO,TONE,TTWO,LONE,LTWO
    2 FORMAT (4F10.5)
    3 FORMAT (5F10.5)
      READ 2,P(1,1),P(1,2),P(2,1),P(2,2)
      READ 2,F(1,1),F(1,2),F(2,1),F(2,2)
      READ 2,R(1,1),R(1,2),R(2,1),R(2,2)
      A = 10.0
      DO 4 I = 1,51
    4 E(1,I) = 0.0
      X(1) = 0.0
      DO 6 L = 1,50
      M = L + 1
    6 X(M) = X(L) + .02
      DO 7 K = 1,51
      PONE(K) = X(K)*(P(1,1)*F(1,1)+P(1,2)*F(2,1))
      PONE(K) = PONE(K)+(1.0-X(K))*(P(2,2)*F(2,1)+P(2,1)*F(1,1))
      PTWO(K) = 1.0 - PONE(K)
      TONE(K)=(X(K)*P(1,1)*F(1,1)+(1.0-X(K))*P(2,1)*F(1,1))/PONE(K)
      TTWO(K)=(X(K)*P(1,1)*F(1,2)+(1.0-X(K))*P(2,1)*F(1,2))/PTWO(K)
      J = 51
   10 IF (TONE(K) - X(J)) 8,9,9
    8 J = J - 1
      IF(J) 11,10,10
    9 LONE(K) = J
   11 J = 51
   14 IF(TTWO(K)-X(J)) 13,12,12
   13 J = J - 1
      IF(J) 7,14,14
   12 LTWO(K) = J
    7 CONTINUE
      PRINT 103,(PONE(J),J = 1,50)
      PRINT 103,(PTWO(J),J = 1,50)
      PRINT 103,(TONE(J),J = 1,50)
      PRINT 103,(TTWO(J),J = 1,50)
      DO 100 N = 2,11
      M = N - 1
      I = LONE(51)
      J = I + 1
      SLOPE = (E(M,J)-E(M,I))/.02
      TEST3 = (TONE(51)-X(I))*SLOPE+E(M,I)
      I = LTWO(51)
      J = I + 1
      SLOPE = (E(M,J)-E(M,I))/.02
      TEST4 = (TTWO(51)-X(I))*SLOPE+E(M,I)
      TEST3 = (-A)+PONE(51)*TEST3+PTWO(51)*TEST4
      DO 99 K = 1,51
      M = N - 1
```

```
      I = LONE(K)
      J = I + 1
      SLOPE = (E(M,J)-E(M,I))/.02
      PAST1 = (TONE(K) - X(I))*SLOPE+E(M,I)
      I = LTWO(K)
      J = I + 1
      SLOPE = (E(M,J)-E(M,I))/.02
      PAST2 = (TTWO(K)-X(I))*SLOPE+E(M,I)
      PAST = PONE(K)*PAST1 + PTWO(K)*PAST2
      TEST1 = X(K)*R(1,1)-(1.0-X(K))*R(1,2)+PAST
      TEST2 = (-X(K))*R(2,1) + (1.0 - X(K))*R(2,2)+ PAST
      IF(TEST1-TEST2) 61,62,62
   61 IF(TEST2 - TEST3) 63,64,64
   62 IF(TEST1 - TEST3) 63,65,65
   63 E(N,K) = TEST3
      GO TO 99
   64 E(N,K) = TEST2
      GO TO 99
   65 E(N,K) = TEST1
   99 CONTINUE
  100 CONTINUE
      DO 101 I = 1,11
      L = I - 1
      PRINT 5, L
    5 FORMAT(27H EXPECTED EARNINGS AT LEVEL,2X,I2 )
  101 PRINT 103,(E(I,J),J = 1,50)
  103 FORMAT(10F10.5)
      CALL EXIT
      END
```

APPENDIX II

COMPUTER SOLUTION FOR THE OPTIMAL STEADY STATE POLICY

In Chapter III an algorithm is presented which allows determination of certain steady state policies associated with partially observable Markov processes. That method is used in this program on the general two state process where the decision-maker has available the options of:

1) estimate underlying Markov state 1 as the current Markov state and act accordingly.
2) estimate underlying Markov state 2 as the current Markov state and act accordingly.
3) use perfect information channel at added cost.

For the two state process the state of knowledge vector, $\underline{s(n)}$, is fully specified by $s_1(n)$ since $s_2(n)=1-s_1(n)$.

The program breaks the continuous $s_1(n)$ into 51 points and determines the optimal policy and associated relative values and gain. The decision is optimal based on the criterion of maximizing expected utility of rewards. Figure 10 of Chapter III very adequately serves as a flow graph of this program.

PROGRAM LISTING - "POLICY"

```
      DIMENSION NA(52),K(51),P(153,51),Q(3,51),V(51),A(51,51),B(51)
      COMMON NA,K,P,Q,V,A,B
C     INITIAL POLICY VECTOR
      NA(52) = 153
      DO 1 I = 1,20
    1 K(I) = 2
      DO 2 I = 21,25
    2 K(I) = 3
      DO 3 I = 26,51
    3 K(I) =1
   27 FORMAT (11F9.5)
   51 FORMAT (6F9.5)
C     FORM Q VECTOR
    4 FORMAT (2F6.3)
   52 FORMAT (7F6.3)
      DO 53 J = 1,51
   53 READ 4, Q(1,J),Q(2,J)
      DO 5 J = 1,51
    5 Q(3,J) = 0.0
   28 FORMAT (2H Q)
C     FORM +P VECTOR
    6 FORMAT (4F6.5)
      DO 7 I = 1,153
      DO 7 J = 1,51
    7 P(I,J) = 0.0
    9 FORMAT (17F4.3)
      DO 200 I = 1,153
  200 READ 9, (P(I,J),J = 1,51)
C     FORM NA(I)
      NA(1) = 0
      DO 10 I = 2,51
      J = I - 1
   10 NA(I) = NA(J) + 3
C     FORM A AND B VECTORS
  102 V(51) = 0.0
      DO 11 I = 1,51
      L = K(I)
      LL = NA(I) + K(I)
      B(I) = Q(L,I)
      A(I,51) = 1.0
      DO 11 J = 1,50
   11 A(I,J) = (-P(LL,J))
      DO 12 I = 1,50
   12 A(I,I) = A(I,I) + 1.0
C     CALL LINEAR EQN SUBROUTINE
      SCALE = 1.0
   50 FORMAT (3F9.5)
      M = XSIMEQF(51,51,1,A,B,SCALE,V)
C     PULL RESULTS
```

```
      GO TO (13,14,14),M
   13 DO 16 I = 1,51
   16 V(I) = A(I,1)
      G = A(51,1)
      V(51) = 0.0
      ITER =1
      DO 100 I = 1,51
      TEMP = -99999.
      NTEMP = 0
      IMIN = NA(I) + 1
      IMAX = NA(I + 1)
      DO 17 M = IMIN,IMAX
      KALT = M - NA(I)
      TEST = Q(KALT,I)
      DO 18 J = 1,51
   18 TEST = TEST + P(M,J)*V(J)
      IF(TEST - TEMP) 17,20,21
   21 NTEMP = KALT
      TEMP = TEST
      GO TO 17
   20 IF(NTEMP - K(I)) 21,17,21
   17 CONTINUE
      IF(NTEMP - K(I)) 22,100,22
   22 ITER = 2
      K(I) = NTEMP
  100 CONTINUE
      GO TO (101,102),ITER
C     PRINT OUTPUT
  101 PRINT 23
   23 FORMAT (16H DECISION VECTOR)
   24 FORMAT (51I2)
      PRINT 24, (K(I),I=1,51)
      PRINT 25
   26 FORMAT (F9.5)
      PRINT 26, G
   25 FORMAT (16H GAIN AND VALUES)
   70 FORMAT (11F9.5)
   71 FORMAT (7F9.5)
      PRINT 70, (V(I),I = 1,11)
      PRINT 70, (V(I),I = 12,22)
      PRINT 70, (V(I),I = 23,33)
      PRINT 70, (V(I),I = 34,44)
      PRINT 71, (V(I),I = 45,51)
      GO TO 103
   14 PRINT 15
   15 FORMAT (23H NO SOLUTION FOR VALUES)
  103 CONTINUE
      CALL EXIT
      END
```

BIBLIOGRAPHY

1) Bharucha-Reid, A.T., Elements of the Theory of Markov Processes and their Applications, McGraw-Hill, 1960.

2) Bellman, Richard E., Dynamic Programming, Princeton, University Press, 1957.

3) Bellman, R., and Dreyfus, S., Applied Dynamic Programming, Princeton, 1962.

4) Drake, Alvin W., Observation of a Markov Process through a Noisy Channel, Sc.D. Thesis, Dept. of Electrical Engineering, M.I.T., May, 1962.

5) Howard, Ronald A., Dynamic Inference, Technical Report No. 10, Operations Research Center, M.I.T., 1964.

6) Howard, Ronald A., Dynamic Programming and Markov Processes, Technology Press, Cambridge, and Wiley, New York, 1960.

7) Kelly, J., "A New Interpretation of Information Rate," Bell System Technical Journal, Vol. 35, pp 917-26, July, 1956.

8) Kramer, J. David M., Partially Observable Markov Process, Sc.D. Thesis, Dept. of Electrical Engineering, M.I.T., February, 1964.

9) Raiffa, H. and Schlaifer, R., Applied Statistical Decision Theory, Division of Research, Harvard Business School, 1961.

10) Schweitzer, Paul, Perturbation Theory and Markovian Decision Processes, Sc.D. Thesis, Dept. of Physics, M.I.T., 1965.

11) Stoopes, G., Study of a Noisy Markov Process, S.M. Thesis, Dept. of Electrical Engineering, M.I.T., June, 1962.

12) Karlin, S., Some Random Walks Arising in Learning Models, Pacific Journal of Mathematics, Vol. 3, 1953.

Security Classification

DOCUMENT CONTROL DATA · R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
MIT Center for Operations Research
77 Massachusetts Avenue
Cambridge, Massachusett 02139

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
Optimum Policies for Partially Observable Markov Systems

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report No. 18 of the MIT Operations Research Center

5. AUTHOR(S) *(Last name, first name, initial)*
Kakalik, James Steven

6. REPORT DATE
October, 1965

7a. TOTAL NO. OF PAGES
76

7b. NO. OF REFS
12

8a. CONTRACT OR GRANT NO.
Nonr-3963 (06) (NR 276-004)

b. PROJECT NO.

c. Contract No.
DA-31-124-ARO-D-209

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Same as 4.

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
--

10. AVAILABILITY/LIMITATION NOTICES
Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES
Sponsoring Military Activity
for 8a: Office of Naval Res.
Methodology Div, Naval Analysis Gp
Washington, D.C. 20360

12. SPONSORING MILITARY ACTIVITY (for 8b)
U.S.Army Research Office
Box CM, Duke Station
Durham, North Carolina

13. ABSTRACT

The problem considered is the monitoring of a discrete-state Markov process through a noisy channel with associated costs. Dynamic programming techniques are used to establish the optimal monitoring decision policies for both transient and steady-state situations. Examples are included.

DD FORM 1 JAN 64 1473

Security Classification

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Dynamic Programming						
Markov Processes						
Optimal Control						
Statistical Decision Theory						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

GPO 886-551